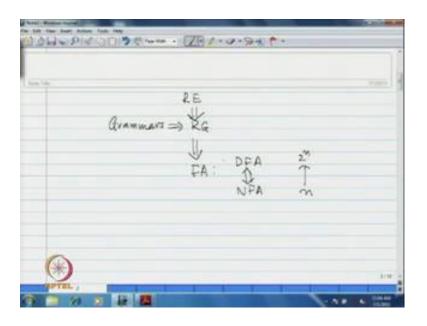
## Formal Languages and Automata Theory Prof. Dr .K.V. Krishna Department of Mathematics Indian Institute of Technology, Guwahati

## Module - 04 Minimization of Finite Automata Lecture - 01 Myhill - Nerode Theorem

Let us first see what the present situation is. In this course, so far we have introduced the following. First, we introduce the notion of formal languages and then we look for the finite representation, in that context we have introduced the notion called regular expressions and the respective languages called regular languages.

(Refer Slide Time: 00:48)



So, after introducing these regular languages, given a language whether it is a regular or not, it is not that easy to understand we understood that point because giving regular expression is not that easy. So, in that context we took the help of grammars and we specialized that to so called right linear grammars, that we are calling regular grammars because our intention is to understand that this regular grammars characterizes regular languages. Of course, we have not proved that we are going to prove all these things. So, through regular expressions whatever that we have defined. So, called regular languages our intention is to show that regular grammars characterizes.

That means given any regular language we can understand that through a regular grammar. Still there we have certain examples of regular languages we are unable to give regular grammar for that. In that context, we gave a tool called finite automaton and through finite automaton we are understanding regular languages. As I hope you understood clearly that giving a regular expression is not that easy and regular grammar is relatively better, but for many of the things I hope now you could have given this finite automaton.

In that we have variants of course,. So, called DFA and NFA. We have observed that these 2 are equivalent. So, here the notion of finite automaton that may be DFA or NFA this assumption lead to capture the information related to regular languages. These characterizes that term. So, in this context when we are proving DFA is equivalent to NFA, the number of states the newer. So, the DFA is introduced and whenever you are given an NFA, if you want to convert into a DFA, what is happening the situation here.

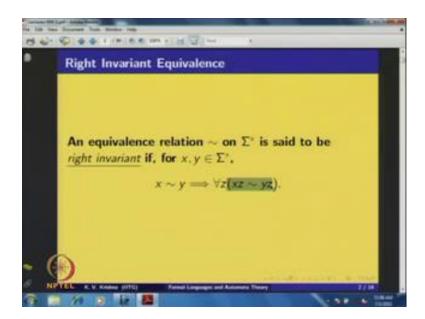
If you have n states here in this conversion you are getting 2 power n states that is exponential number of states. Now, in order to understand that what is the possible minimal DFA or you know when you are considering a particular language corresponding to that, if you are if you are constructing a d f an NFA and converting to DFA use, is it the situation that you will always have that many number of states or whether there are any states which are useless.

So, that kind of aspects now in this module we are going to discuss. So, here our concentration is to understand that what is the minimal DFA to accept a regular language, that is our concentration now. So, in this cycle of course, once again when to conclude here this particular cycle, this finite automata. We are going to observe that in the following lectures that all these are equivalent.

So, the present concentration of course, at present we are concentrating to understand that what is the minimal DFA to accept a regular language. So, the present lecture is essentially present module essentially concentrating on that 1 or 2 lectures in this direction. In this direction we require one important tool called Myhill Nerode theorem, that I will be discussing in this lecture. In fact this a very important characterization for the languages accepted by DFA. Of course, eventually what we are understanding the language accepted by DFA is regular languages. So, this is very important

characterization for regular languages. So, in this lecture I will concentrate on. So, called Myhill Nerode theorem and to introduce that theorem I require some basic definitions in that direction.

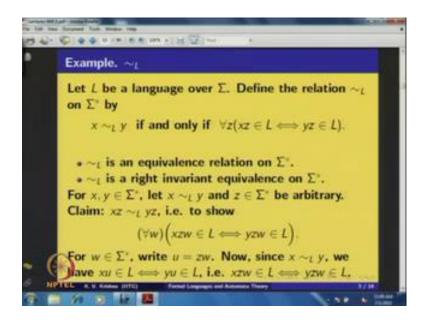
(Refer Slide Time: 04:58)



First let me introduce this, an equivalence relation tilde on sigma star like strings over an alphabet sigma is said to be right invariant, if you take any 2 strings x y, if they are related with respect to tilde, then any string you take z and concatenate it and right side of both the strings. Then there should also be related with respect to tilde. So, if this property satisfied for all strings x y which are related, if you concatenate any string z on their right hand side the resultant string should also be equivalent with respect to that tilde.

So, such an equivalence relation we call it as right invariant equivalence relation. Let us look at an example. So, that you can understand this concept right better consider. So, this example I am defining tilde L. Of course that I will be using throughout this lecture.

(Refer Slide Time: 05:55)



So, please remember now, you consider a language over sigma that means it is a subset of sigma star. Define the relation tilde L on sigma star with this condition, that x related to if you take any 2 strings x and y in sigma star, you relate them if and only if, for any string z x z is in L if and only if y z is in L. So, whenever the x z is in L, y z should also be in L. So, this and vice versa of course.

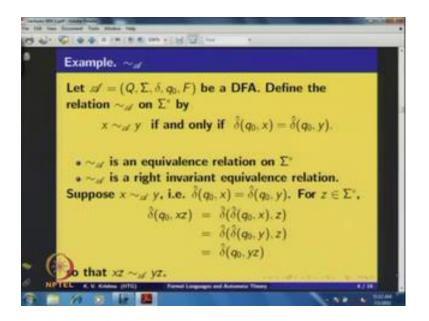
So, with this condition we are defining this tilde L. Of course, we can quickly understand that this is an equivalence relation because to understand a relation is an equivalence you have to understand reflexivity, symmetry and transitive. So, any string x is related to itself because if you concatenate any string z, the resultant string and both the situations here are same and therefore, both are in L or not in L.

Thus, you can quickly understand that it is reflexive. Symmetry follows very quickly because of this if and only if condition here and transitive to also you can verify to understand that this tilde is an equivalence relation on sigma star. Now, our point here of course, important thing you have to cross check here in this context is right invariant property. That means x related to y implies x z related to y z for all z that is what we have to understand for right invariant relation.

So, take x y in sigma star and choose takes x y in sigma star and assume they are related and pick up any z arbitrary in from sigma star. Now, we have to show that x z related to y z with respect to this relation, that is we have to show that for all w in sigma star x z w

is in L if and only if y z w is in L. This is the condition we have to cross check. Now, if you write for any w take an arbitrary w in sigma star and write u to be z w. Now, you see that since x related to y with respect to tilde L what we have x u related x u is in L if and only if y u is in L this is the definition of tilde L. So, x u is in L if and only if y u is in L and what is u, we have chosen that is z w t. That means what are the property that we are aiming to that is this we have got it and that is this x z w is in L if and only y z w is in L. Hence this tilde L is a right invariant equivalence relation. Now, I give you one more example.

(Refer Slide Time: 09:08)



This is also very important in this lecture, that is I am defining it as tilde a for a DFA a. Consider a DFA q sigma delta q naught f, take a DFA. Now, define the relation tilde a on sigma star as below, x and y are related with respect to tilde a if and only if, you put those strings in the initial state, you should essentially reach to the same state. If you are reaching to the same state, then you say they are related. You can understand again that this is an equivalence relation because if you take any string that you related itself because what are the state that you are reaching to with respect to x of course, that is fixed in a DFA and hence x related to x for all x.

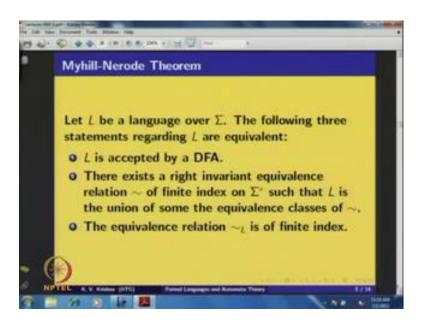
Symmetry is very straight forward because here we have we are putting equal to. So, x is reaching to a particular state and you know y is reaching to a particular state and they are same then of course, vice versa. Hence x related to y implies y related to x. When you

are looking for the transitivity, because of the equal to here you are transitivity also very quickly and hence this is an equivalence relation tilde a is an equivalence relation.

So, here what we have to understand that it is a right invariant equivalence relation let us see. Take x related to y that is this property satisfy. Now you pick-up any arbitrary z in sigma star, what I have to observe that delta cap q naught x z is equal to delta cap of q naught y z, that is what we have to observe, to show that x z is related to y z.

Now, consider this delta cap q naught x z. I hope by know you have proved that this is equal to this you can use induction to prove this property. Now, since these two are equal you can take in place of delta cap of x naught, q naught delta x. You can replace it with delta cap q naught at y. Again using the property you can have this. Thus x z you related to y z for all z, let us we have that is an arbitrary thing. Thus, you can understand this is also a right invariant equivalence relation on sigma star. Of course, and sigma star tilde a is defined on sigma star.

(Refer Slide Time: 11:40)

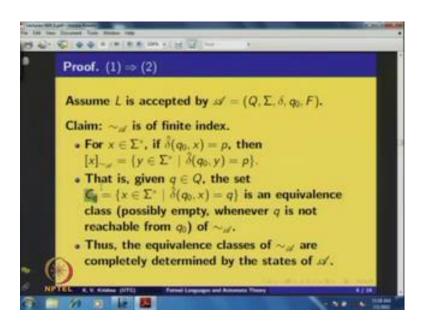


Now, let me state Myhill Nerode theorem. Let L be a language over sigma, the following three statements regarding L are equivalent, what are those statements. Statement 1 L is accepted by a DFA. Of course, once eventually if you prove that the finite automata is a character, it captures the properties of regular languages that means it is it is characterizing regular languages. Then this theorem use a characterization for regular languages as I had mentioned.

What is the point here is, there exist a right invariant equivalence relation tilde a finite index on sigma star such that, L is the union of some more equivalence classes of tilde. So, what we are trying to say here is the second point, you can find a right invariant equivalence relation tilde such that, L is and of course, it is the finite index such that, L is union of some of its equivalence classes.

Number three is the equivalence relation tilde L as defined just now with respect to L because given a language L you can talk about tilde L as just we have defined. The equivalence relation tilde L is a finite index. So, this theorem claims that these 3 are equivalent. L is actually DFA and there is a right invariant equivalence relation tilde finite index, such that L is union of some of its equivalence classes.

(Refer Slide Time: 13:23)



The equivalence relation tilde L is a finite index, these 3 are equivalent. We prove 1 implies 2 and 2 implies 3 and 3 implies 1 to show that these 3 are equivalent. To show 1 implies 2 assume L is accepted by a DFA say a to b this, q sigma delta q naught f. First what we will show, that in point number 2 what we have to show there is a right invariant equivalence relation, which is the finite index and L is union of some of it equivalence classes, this is the point number 2 we have to prove.

Here in this context what we will observe that that tilde into that is, you know we have tilde a will satisfy the tilde in 2, that means what we have to observe that, tilde is the finite index because we have already observed that tilde a, is a right invariant

equivalence relation on sigma star. So, what we have to observe the remaining 2 points, one is it is a finite index.

Number 2 is L is union of some of its equivalence classes. These two points we have to observe. So, tilde a you know already the claim is it is a finite index. So, to show it is a finite index. First let me observe the following 2 points, for x and sigma star delta cap of q naught x, if it is equal to p, then the equivalence class containing x is all those strings in sigma star, if you put them in initial state.

If you reach to p, that is equivalent to x because if you remember the definition of tilde a 2 strings with respect to tilde a. They are equivalent if you put those 2 strings in the initial state of DFA here that is in q naught you should reach to the same state then we say that those two are equivalent. That means if you take any string x what are the strings equivalent to x, that is the equivalence class containing x is precisely.

Although strings in sigma star if you put them in an initial state q naught if you reached to the state p what is be here, the state that you are reaching via x from the initial state q naught. This is what is precisely the equivalence class containing x, this is the point number 1. Point number 2 because here the state to which you are reaching that is characterizing the equivalence class. Now, corresponding to each state, now let us look at all those strings.

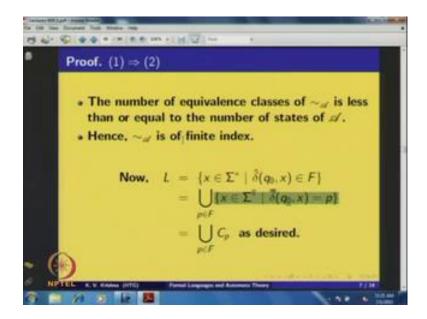
Let me call for a given q in q for a given state q consider c q the class of q I am calling is although strings if you put them in the initial state q naught if you raise to the state q let me call it as c q. This is an equivalence class of course, this can be empty, if no string can reach to this state q because if that is not reachable y a, using any string then of course, this is empty let me consider empty also, in this context.

So, what I am saying here, this is an equivalence class of tilde a because what are all those strings, if we put in the initial state to reach to q, they are all equivalent to each other. This forms an equivalence class. Thus what you can understand equivalence classes of tilde a are completely determined by the states of a because from the point number one, you understood that the state is coming into the picture.

Now, take every state and said the class c q and collect all those strings, this can be empty if that particular state is not reachable from the initial state, but wherever it is non

empty all those strings which are reaching from the initial state with that particular state, they are all equivalent. That is an equivalence class with respect to this tilde a. From these 2 points what you can understand that, this tilde a is completely determined by the states of a. Moreover the number of equivalence classes of tilde a is less than or equal to the number of states of a.

(Refer Slide Time: 18:00)

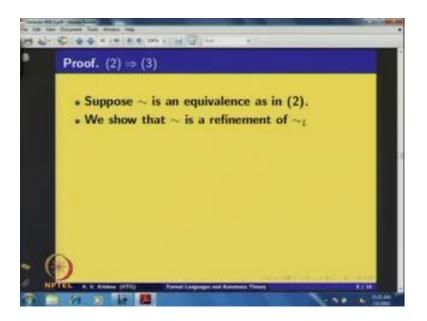


The reason why then I have every state gives an equivalence class to you of course, possibly empty in whenever the state is not reachable then equivalence class what I am calling here is empty set. So, thus the number corresponding to each state you get an equivalence class possibly an empty class some of the cases. So, thus the number of equivalence classes of tilde a is less than or equal to the number of states of a.

Hence tilde is a finite index because a has only finite number of states. Therefore, tilde a, is a finite index. Of course, here index we mean index of an equivalence relation you mean the number of equivalence classes of that relation. Now, we want to prove that the second point L is union of some of it equivalence classes. Here this tilde a is equivalence classes. Now, consider this L by definition this is, although the x n sigma star if you put them in the initial state you will reaching to the 2 1 of the final states. That means delta cap of q naught x is in f that is union of these sets where you reach to a particular final state because for all final states I am considering these sets.

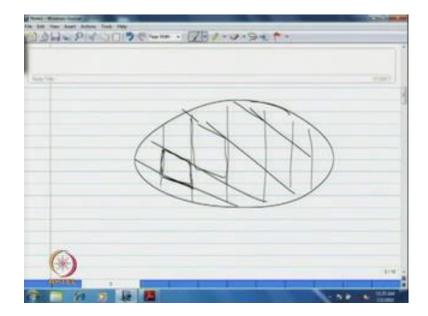
Some of the non-final some of the final states may not be reachable in which case this set is empty. Now, just we have proved that this set is an equivalence class we are calling it as c p corresponding to a state p. Thus you understand that L is union of c p for p belongs to f as desired. Thus we have observed that 1 implies 2 that means we could identify a right invariant equivalence relation. Here that is tilde a, which is the finite index number 1 and number 2 L is the language L is union of some of its equivalence classes. We have proved 1 implies 2. Now, let us consider 2 implies 3. To prove 2 implies 3 what we have to observe that tilde L is a finite index the number of equivalence classes of tilde L is finite.

(Refer Slide Time: 20:43)



Suppose tilde is an equivalence as into that means tilde is a right invariant equivalence relation of finite index number 1 and number 2 is L is union of some of this equivalence classes. So, that is what I am assuming tilde is an equivalence as in 2 point number 2 of the theorem. We show that tilde is a refinement of tilde L, what is refinement. If you suppose this is the set under consideration and if you consider equivalence.

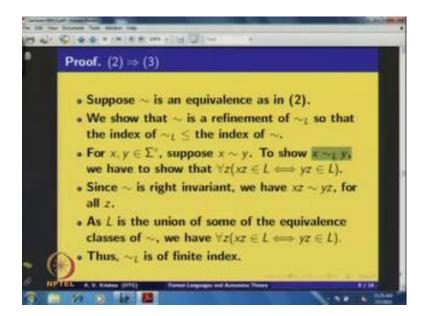
(Refer Slide Time: 21:26)



Relation equivalence relation, you know equivalence relation partitions the set. Now, if you further refine this relation. That means if you break this equivalence classes further. For example, may be we are breaking this direction or whatever the classes. Here I am getting, we can clearly see that what are the partition that I am getting here each equivalence class that means equivalence class here I mean, the portions that are shown with a boundaries of this cross lines with the vertical lines. So, this portion is this is the new equivalence class.

For example here, this is a new equivalence class. These equivalence classes are contained in the original equivalence classes because there subsets of the because we are partitioning, further partitioning this. Now, we say this is the refinement, we are further refining this equivalence classes. So, that means if you take any 2 elements in this set, if they are related with respect to new equivalence relation, they are related to the original with respect to original relation because new equivalence class is a subset of original equivalence class. So, what we are trying to show that, this tilde is a refinement of tilde L.

(Refer Slide Time: 23:00)



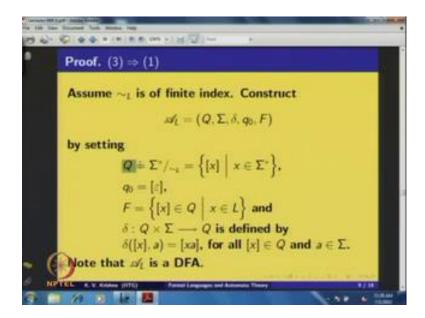
That means once we show that this is a refinement, as I have explained that the number of equivalence classes of tilde is more than the number of I mean more or equal to the number of equivalence classes of tilde L. That means an index of tilde is greater than or equal to index of tilde L. As it is given that tilde is a finite index, tilde L is also is the finite index. So, once we show that tilde is a refinement of tilde L, then we are through.

Now, suppose as I have mentioned that if x related to y with respect to tilde, we have to show that x related to with respect to tilde L also, to observe that this is tilde is a refinement of tilde L. Now, for that purpose for x y in sigma star, suppose x related to y with respect to tilde to show x related to with y with respect to tilde L, what we have to show, for all that if you concatenate z from its right side this is in L if and only if, y z is in L.

Now, since tilde is right invariant that is what is hypothesis given in 2, the point number 2 are the theorem. So, tilde is right invariant what we have x z related to y z for all that. In point number 2 we have this information also, L is union of some its equivalence classes some of the equivalence classes of tilde that means whenever x z related to y z that means x z is in L, then y z should also be in L. So, that is we have for all z x z is in L if and only if y z is in L that is what is the desired property we look for, to show that x related to y with respect to tilde L. Hence what we have observed, x related to y with

respect to tilde implies x related to with respect to tilde L also. Hence tilde is a refinement of tilde and thus tilde L is a finite index. So, this is gives you 2 implies 3.

(Refer Slide Time: 25:04)



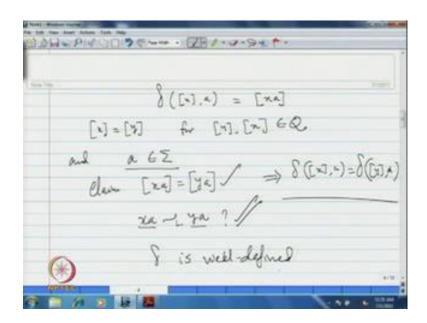
Now, we prove 3 implies 1. In 3 what we have, tilde L is the finite index let us assume that and what we have to show. We have to show that there is a DFA to accept L, L is accepted by some DFA. Now, we construct the DFA in this portion 3 implies 1. Now, construct A L by taking q sigma delta q naught f. Of course, here I have to give you that q the states at you consider it as sigma star by tilde L, that means the partition of sigma star with respect to the equivalence relation tilde L.

Hence all the equivalence classes bracket x, x in sigma star that is what we consider it as state set in A L and what is the initial state. You consider the equivalence class containing epsilon, the empty string. Then you consider a final states or as F all those equivalence classes with which are in you know if you take any string in L, the equivalence class containing that x you consider as a final state. Now, you have to give a transition function in case of DFA.

Let me define delta from Q cross sigma to Q defined by delta of bracket x at a, we assign it to bracket x a, the equivalence class containing the string x a. Of course, for all states bracket x and for all a in sigma. We have to understand that this is a DFA. To understand this is DFA what do, you require these of course, sigma is given to the alphabet that is a finite set alphabet means I am what I have to understand that q is a finite set number 1.

Number 2 this delta is a function from Q cross sigma to Q. That means we have look at we have to prove the well defines of delta. So, that this is a DFA. So, to observe these 2 points of course, since tilde L is a finite index you can quickly see that the number of equivalence classes of tilde L is finite. Thus the partition, that means the number of equivalence classes here is finite. So, Q is a finite set very quickly that Q is a finite set.

(Refer Slide Time: 28:23)



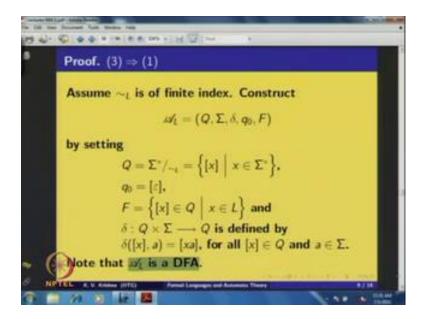
To understand that this delta is well defined. How is delta defined, this is defined as to show this delta is well defined consider bracket x equal to bracket y. Consider two states for bracket y bracket x in q. These are the states consider arbitrary a in sigma. The claim is bracket x a is equal to bracket y a because what we have to show delta of bracket x at a is equal to delta of bracket y at a. That is what we have to show that bracket x a is equal to bracket y a.

So, that means what we have show with respect to, show these 2 equivalence classes are same we have to show that this x a with respect to tilde L it is related to y a, that is what we have to show. What is tilde a, you know already it is a right invariant equivalence relation. That means, if you concatenate any string on right hand side, the resultant strings are also equivalent.

Here a is in sigma, that is a string you can of length 1. If you concatenate a to x as well as y, they should also be related with respect to tilde a because tilde L is a right invariant equivalence relation. Hence you have this property, once you have this property, these 2

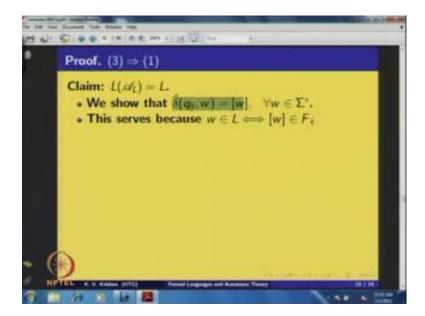
strings are in the same equivalence class. That means the equivalence classes containing x a and y a are same and does what you have, delta of x bracket x at a is equal to delta of bracket y at a. Thus, you can understand this delta is well defined, this is well defined.

(Refer Slide Time: 30:40)

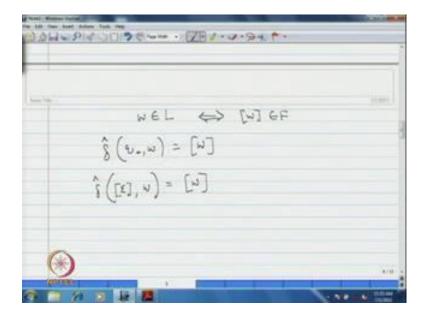


So, once you understand that this delta is a map from Q cross sigma to Q this with this assignment, we have this A L is a DFA deterministic automaton. So, we constructed DFA. We proved that the language accepted this DFA is L the claim is L of A L is equal to L.

(Refer Slide Time: 30:49)



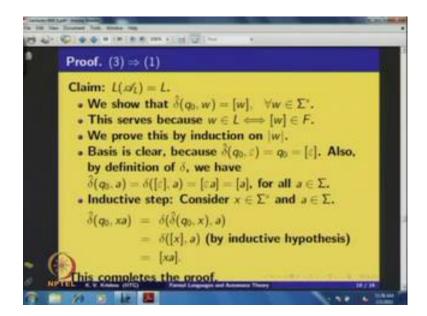
(Refer Slide Time: 30:58)



For that purpose, we show that delta cap of q naught w is equal to bracket w. So, the state, the equivalence class the state the resultant state will be the equivalence class containing w. This serves the purpose, if you show this, this point this is sufficient because if you take any string w is in L, that if and only if bracket w is in F, the reason why w is in L if and only if, bracket w is in F.

How this is, once we have proved this thing delta cap of q naught w equal to bracket w, once we prove this the initial state here of course, is delta cap of we are proving this that is what we are proving. If the state is in final state then w is in f that is straight forward if this is w is in L Whenever w is in L what is the requirement, whenever you put this w in the initial state you should you should reach to a state that is in F.

(Refer Slide Time: 32:41)



So, with this criteria delta cap of q naught w is equal to bracket w. If you can prove then w is in L if and only if bracket w is in f is clear. Thus what will happen the language accepted by A L is equal to L. We prove this assertion by induction on the length of w. So, induction basis is clear for strings of length 0 because if you take a string of length 0, that is epsilon. If you put in this that is what is q naught by definition of delta cap of course, in any DFA.

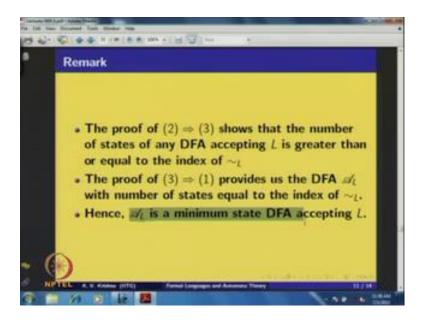
Here q naught is bracket epsilon thus we have, the basis of the induction that is for those strings of length 0 of course, empty string more over you can in fact observe that by definition of this delta this assertion is true for those strings of length 1. Thus delta we have defined with this condition. Now, if you consider delta cap of q naught at a, for that strings of length a, that means there is a delta cap is nothing as, but delta and delta is defined as...

So, and this string is nothing as, but a epsilon a is nothing as, but a that is you are getting class containing a. Hence what are assertion we have that is true for all those strings of length 0 and all those strings of length 1. So, induction basis from this point you can observe. For inductive step you consider a string in sigma star some x and consider a in sigma what I have to observe the delta cap of q naught x a is equal to bracket x a, that is what we have to prove.

So, consider delta cap of q naught x a this equal to this. That is the definition of delta cap, by induction hypothesis we have this delta cap of q naught at x is bracket x, that is delta of bracket x at a, by definition of delta you know this is bracket x a. Thus what we have delta cap of q naught x a is equal to bracket x a.

Hence by induction what we have delta cap of q naught at w is equal to bracket w. For all w in sigma star and hence any string is in L. If you put it in the initial state of this particular automaton, you will be reaching to a final state precisely the equivalence class containing the string. Thus the language accepted by this automaton a L is L. Hence, L is accepted by some DFA of course, here A L this completes the proof of Myhill Nerode theorem. So, we have proved that 1 implies 2 two implies 3 and 3 implies 1. Thus, if the condition the statements 1, 2 and 3 are equivalent. Now, let me give you a remark.

(Refer Slide Time: 36:16)



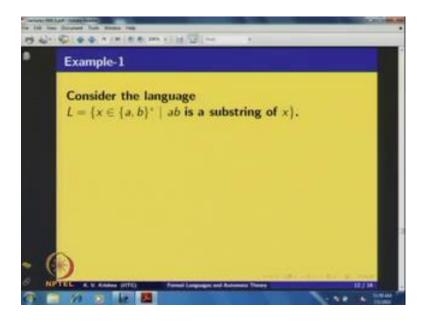
The proof of 2 implies 3 shows that the number of states of any DFA accepting L is greater than or equal to index of tilde L. Given a language L, considering DFA accepting L in 2 implies 3, what we have proved the number of states of that DFA a is greater than or equal to the index of tilde L.

That is what we have proved because to observe that tilde L is a finite index we have proved this point in 2 implies 3. In the proof 3 implies 1 we have provided DFA namely A L, if the number of states equal to the index of tilde L because in that A L the states are essentially the equivalence classes of tilde L. So, that means the states what we have

considered there equivalence classes there is the number of equivalence classes equal to the number of states for that A L.

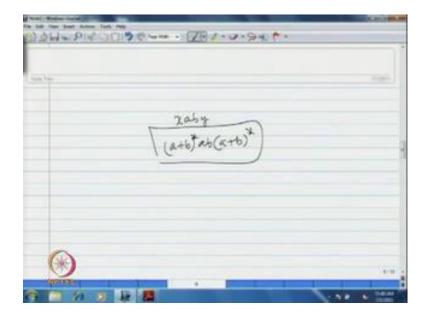
So, from these 2 points we can understand that A L is a minimum state DFA accepting L. So, given L you know what is tilde L, from tilde L you know the equivalence classes from the equivalence making those equivalence classes as states, we have constructed A L. Since any DFA accepting L should have the number of states of any DFA accepting L to the more or equal states then the index of tilde L and A L is, in fact having the number of states equal to the index of tilde L. We can conclude that a L is minimum state DFA accepting L. So, we have from Myhill Nerode theorem we have a minimum state DFA accepting L that is what is the conclusion at this point of time.

(Refer Slide Time: 38:18)



Now, let me give one or two examples as applications of Myhill Nerode theorem. Thus Myhill Nerode theorem is characterizing, once again I am observing that Myhill Nerode theorem is characterizing, the languages that are accepted by DFA. Eventually what I have we point out that Myhill Nerode theorem characterizing regular languages. So, let me consider this example, you know this is a regular language, you can give a regular expression for this. Consider the language x in consider the language with all those strings having a b as substring, what is the regular expression for this.

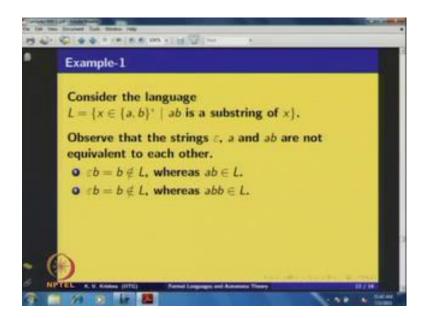
(Refer Slide Time: 39:18)



Any string is here is of the form x a b y for any x and y in over a b. Thus the regular expression for this is, a plus b star a b a plus b star, this you know and hence this is a regular language. You know a DFA accepting this language also that we have already discussed have constructed. From Myhill Nerode theorem, we should understand that the tilde L should be a finite index.

So, let us calculate the number of equivalence class of tilde L in this example. To understand using Myhill Nerode theorem this is a regular, I mean this is a language accepted a DFA. First observe that the strings epsilon a and a b are not equivalent to each other. Let us observe these point. So, to show epsilon a are not equivalent to each other, we have to find a string, such that if you concatenate that string on the right to it, right hand side of epsilon and to a one string should be in L, one string should not be in other string should not be in L, that is how we have to find.

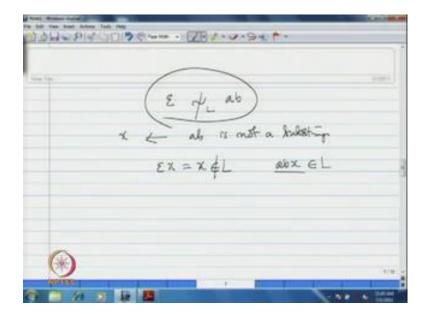
(Refer Slide Time: 40:49)



Now, in particular suppose if you consider b, the string b epsilon b is b. You know b is not in l, but if you if you concatenate b to a on its right hand side that is a b that is in L. So, you can quickly observe from this that, a and epsilon are not equivalent because we could identify a string b here that, concatenating b to the right hand side of both the strings one string you are getting in L other string you are not getting in L.

Thus a and epsilon are not equivalent with respect to tilde L. Similarly, we observe that this epsilon and a b are not equivalent. If you consider any string, which is not containing a b, if you concatenate that to epsilon right hand side, you will get of course, L definitely in the same string and thus that is not in L. If you concatenate that string any string to a b on right hand side anyway that is having a b and thus that is a string in L. So, you get strings 1 is in L other is not in L. So, that you can understand that epsilon and a b are not equivalent. So, what I am trying to say here is to show a and x a b are not equivalent with respect to tilde L.

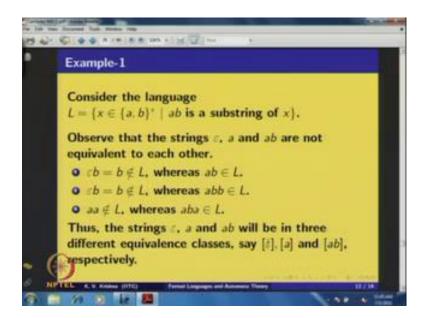
(Refer Slide Time: 42:21)



You choose any string x in which a b is not a substring, for which a b is not a substring. Suppose, if you consider this concatenate x to epsilon right hand side. This is resultant string is x only and if you concatenate x to a b on the right hand side resultant string anyway is having a b a substring. You see this string x which is epsilon x is not having a b a substring. So, that is not in L whereas this string is always in L, as a b is substring of this a resultant string.

Hence, we observe that epsilon and a b are not equivalent. Any string x I general string I have mentioned, in particular here we are considering b. If you choose b, b is not having a b as substring and you can understand that epsilon and a b are not equivalent. Now, we also observe that a and a b are not equivalent.

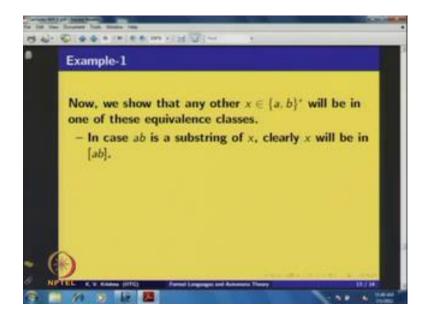
(Refer Slide Time: 43:30)



Here, I have taken the string a. If you concatenate a to the right side then of course, the resultant string is a, a that is not in L. If you concatenate a to a b at right hand side that is a anyway string of L. Thus a when this little a distinguishes a and a b and therefore, a and a b are not equivalent because to show two strings are equivalent with respect to tilde L. If you pick up any string from sigma star concatenating that string right side of both the strings the resultant strings both should be in L or should not be in L.

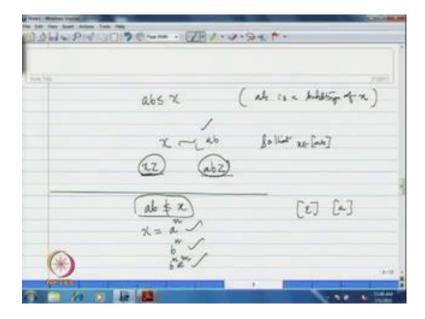
So, with that criteria we have cross check and from these three points. I hope you can pick you understood that, these three are distinguishable to each other. Thus, what is the conclusion here, these three strings should be in different equivalence classes. So, let me say epsilon is in the equivalence class bracket containing epsilon. Let me call bracket a for the equivalence class containing a bracket a b for this. So, let me assume these three classes not there from this. Now, what we are going to show that, if you take any other string in sigma star that string should be in one of these equivalence classes.

(Refer Slide Time: 45:02)



Thus, precisely we will have these three equivalence classes. Hence, we can understand that tilde L is a finite index. Let me observe this point also, pick up a string x. Now, I see the property of x in two cases whether a b is a substring of x or a b is not a substring of x. If a b is a substring of x then x clearly will be.

(Refer Slide Time: 45:34)



Suppose, a b is a substring of x. Let me use this notation a b less than equal to x I mean a b is a substring of x, so that what we are a trying to say that x is related to a b. So, that this x will be in bracket a b. This is very quick because to x if you concatenate any string

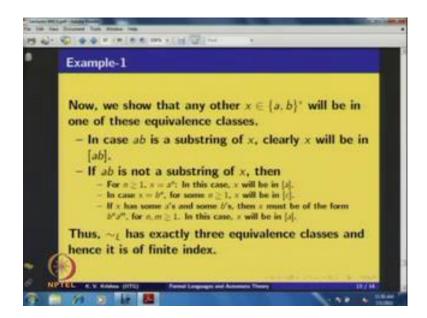
z whether it has a b or not a having a b. Since, already there is a b in x in the resultant string x that you will have a b a substring.

Since, with the say with the same property if you concatenate that z with to a b this a resultant string is also having a b a substring. Thus this is in L and this is also in L irrespective of whatever that z we are concatenating. Hence, these two are equivalent x and a b are equivalent. So, if you take any string x in a b star. If it is having a b a substring that clearly will be in bracket a b. Now, if you consider the other case that if a b is not a substring of x then what are the possibilities.

If a b is not a substring of x, then let me see x y be of the form a power n or by b power n in which a b is not there or when you some a's and b's. Since, a b should not be there you can have some number of b's followed by some number of a's, but of course, after that you should not have any b's after a's, because if you have any b after follow following to this a's, then you will always get a b a substring.

Thus I can categorize those strings x for which a b is not a substring means it may be a power n may be b power n, there is also form b power n or b power n a power n these 3 cases. Now, in each case we can discuss that these strings will fall in some of the equivalence classes among bracket epsilon bracket a. Of course, whether it will be in bracket a b or not we will observe here. For n greater than equal to 1, because if it is empty string that is 1 situation that I will consider in the case of b power n for n greater than or equal to 0, here at least one a for n greater than equal to 1.

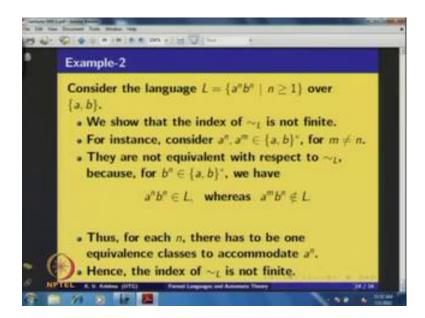
(Refer Slide Time: 48:36)



The other case I will consider. So, for n greater than equal to 1. If you consider x to be of the form a power n in this case x will be in bracket a as I have discussed. So, far you can take it to an exercise and observe that x will be in bracket a, in this case and if you take x which is of the form b power n.

Of course, n greater than equal to 0 also you can put where in which case that is epsilon. That will be in bracket epsilon and if x has some a's and b's as I had mentioned x must be of the form this b power n a power n for m n greater than equal to 1. In this case you can understand that, this will be in bracket a. Thus this tilde L has exactly 3 equivalence classes. Hence, it is a finite index and the conclusion from Myhill Nerode theorem that this language can be accepted by a DFA of course, you know already that this language, DFA accepting this language. So, let me consider one more example. We know this is language a power n b power n for n greater than or equal to 1.

(Refer Slide Time: 49:51)



So, far we have not constructed any of course, we have a grammar for this, but not a regular grammar we have constructed a conditioner grammar for this. In this example, we observe that tilde L for this language L a power n b power n, n greater than equal to 1. We observe that tilde L is of the index of tilde L is infinite. It is not finite and does using Myhill Nerode theorem, you can say this cannot be accepted by any DFA. As we are saying that DFA characterizes the properties of regular languages. Of course, from this we can conclude that a power n, the language a power n b power n, n greater than equal to 1 is not a regular language.

So, let me observe this point. We show that the index of tilde L is not finite. For instance you consider these two strings a power n, b power m these two strings from sigma star. Here sigma is a b with m different from n consider these 2 strings they are not equivalent to they are not equivalent to be equivalent with respect to tilde L, because if you consider b power n. If you concatenate b power n to a power n on its right hand side that is an element of L. Whereas since m different from n a power m b power m is not an element of L. Thus for each n there has to be one equivalence class to accommodate a power n, the string a power n. Thus this tilde L is not a finite index. Therefore, the conclusion from Myhill Nerode theorem is that there cannot be any DFA accepting this language.