

Theory of Computation
Professor Subrahmanyam Kalyanasundaram
Department of Computer Science and Engineering
Indian Institute of Technology Hyderabad
Proving the Myhill-Nerode Theorem

(Refer Slide Time: 0:16)

Myhill-Nerode Theorem

(John Myhill and Avni Nerode, 1958)

Not in the book. Only stated as exercise 1.91 and 1.92.

This provides a necessary and sufficient condition for a language to be regular.

Def 1: Let x, y be strings over Σ and L be a language over Σ . We say that x, y are distinguishable by L if $\exists z \in \Sigma^*$ such that $xz \in L$ and $yz \notin L$ or vice versa. ($xz \notin L$ and $yz \in L$)

If x, y are not distinguishable by L , we say that

over Σ . We say that x, y are distinguishable by L if $\exists z \in \Sigma^*$ such that $xz \in L$ and $yz \notin L$ or vice versa. ($xz \notin L$ and $yz \in L$)

If x, y are not distinguishable by L , we say that they are indistinguishable by L , and denote it by $x \equiv_L y$.

Exercise 1: Show that \equiv_L is an equivalence relation.

- (1) Reflexive: $x \equiv_L x$
- (2) Symmetric: $x \equiv_L y \Rightarrow y \equiv_L x$
- (3) Transitive: $x \equiv_L y$, and $y \equiv_L z \Rightarrow x \equiv_L z$

This implies that the relation \equiv_L partitions Σ^* into equivalence classes.



into equivalence classes.

Example: mod 5 equivalence relation partitions all integers into 5 equivalence classes - based on the remainder when divided by 5.

Similarly \equiv_L relation partitions Σ^* into equi. classes.

Def 2: Let L be a language and X be a set of strings. X is pairwise distinguishable by L if every two distinct strings $x, y \in X$ are distinguishable by L .

Def 3: The index of L is the size of the largest set X of strings such that X is pairwise distinguishable by L .

Def 3: The index of L is the size of the largest set X of strings such that X is pairwise distinguishable by L .

In other words, index of $L =$ No. of equivalence classes of Σ^* as determined by \equiv_L .

Myhill-Nerode Theorem: A language L is regular iff (if and only if) it has a finite index.

Moreover, the index of L is equal to the size (no. of states) of a smallest DFA that recognizes L .



Hello and welcome to lecture 15 of the course Theory of Computation. In lecture 14 we stated Myhill-Nerode theorem which was a necessary and sufficient condition to show for languages being regular. We set up definitions and then we stated the theorem.

(Refer Slide Time: 0:35)

Myhill-Nasade Theorem: A language L is regular iff (if and only if) it has a finite index.
Moreover, the index of L is equal to the size (no. of states) of a smallest DFA that recognizes L .

Lemma 1: If L is recognized by a DFA with k states, then $\text{index}(L) \leq k$.

Lemma 2: If $\text{index}(L) = k < \infty$, then there exists a DFA with k states that recognizes L .

Proof of Myhill-Nasade theorem assuming lemmas

(\Rightarrow) Suppose L is regular. Then there is a DFA that recognizes L . Consider a smallest DFA that recognizes L . Let this DFA be M , and let M have k states. By lemma 1, we have $\text{index}(L) \leq k$.

$$\text{index}(L) \leq \text{Size of the smallest DFA that recognizes } L.$$

(\Leftarrow) Suppose L has finite index, say k . By lemma 2, there exists a DFA with k states that recognizes L . So L is regular.

(\Leftarrow) Suppose L has finite index, say k . By lemma 2, there exists a DFA with k states that recognizes L . So L is regular.

$$\left. \begin{array}{l} \text{Size of the smallest DFA} \\ \text{that recognizes } L \end{array} \right\} \leq \text{index}(L)$$

Combining, we get

$$\text{index}(L) = \text{Size of the smallest DFA that recognizes } L.$$

Notation: $\delta^*(q, x)$ for $x \in \Sigma^*$ denotes the state reached by the DFA starting from q and reading



The theorem states that a language L is regular if and only if it has a finite index and also this index is also the size of the smallest DFA that recognizes that language, if it is finite of course. We said that the proof boils down into two lemmas, basically either direction of the implication of the proof. We quickly saw how these two lemmas imply the theorem which was fairly straightforward.

(Refer Slide Time: 1:14)

that recognizes L .

$\delta(q, a) = r$ $(q) \xrightarrow{a} (r) \xrightarrow{b} (s)$ $\delta^*(q, ab) = s$

Notation: $\delta^*(q, x)$ for $x \in \Sigma^*$ denotes the state reached by the DFA starting from q and reading the string x .

lemma 1: If L is recognized by a DFA with k states, then $\text{index}(L) \leq k$.

Proof: We will show that any two strings that end in the same state are indistinguishable.

Suppose L is recognized by a DFA M with k states.
Suppose, for the sake of contradiction that $\text{index}(L) > k$.



And now what remains is to show these lemmas. So, now let us proceed to the proofs of the lemmas 1 and 2. So, before stating the lemmas, I will just set up a brief notation. So, we have the definition of the transition function $\delta(q, a)$. Suppose this is equal to r . This means that from state q , if you see the symbol a you reach state r . If you see the symbol a where a is a symbol of the alphabet.

Now here I am defining sort of a shorthand notation $\delta^*(q, x)$ is exactly like this but instead of a symbol from the alphabet I am defining it for a string, not a symbol. For instance, x could be ab so suppose now if you reach if you read ab you get to state s . So, if you read ab from q you go to r and then go to s so then you will say $\delta^*(q, ab) = s$.

The δ^* means that the number of steps could be more than one then or the length of the string is equal to the number of steps in a DFA which could be more than one. So, when you start from q and read the string where you end up this is δ^* . This will be used in our proof so that is why I am stating this notation up front.

(Refer Slide Time: 2:59)

Lemma 1: If L is recognized by a DFA with k states, then $\text{index}(L) \leq k$.

Proof: We will show that any two strings that end in the same state are indistinguishable.

Suppose L is recognized by a DFA M with k states. Suppose, for the sake of contradiction that $\text{index}(L) > k$.

This means there exists X such that X is pairwise distinguishable by L , and $|X| > k$.

Let q_0 be the starting state of M . By pigeonhole principle, there exists two strings $x, y \in X$, $x \neq y$ such that $\delta^*(q_0, x) = \delta^*(q_0, y)$.

Let q_0 be the starting state of M . By pigeonhole principle, there exists two strings $x, y \in X$, $x \neq y$ such that $\delta^*(q_0, x) = \delta^*(q_0, y)$ (x and y end in the same state).

For any $z \in \Sigma^*$, we now have

$$\begin{aligned} \delta^*(q_0, xz) &= \delta^*(\delta^*(q_0, x), z) \\ &= \delta^*(q_0, z) \\ &= \delta^*(\delta^*(q_0, y), z) \\ &= \delta^*(q_0, yz) \end{aligned}$$

So $xz \in L \Leftrightarrow yz \in L$. So x, y are indistinguishable by L . So $x \equiv_L y$. This is a contradiction. So $\text{index}(L) \leq k$.

(John McMillan and David D. Reid, 1958)

Not in the book. Only stated as exercise 1.91 and 1.92.

This provides a necessary and sufficient condition for a language to be regular.

Def 1: Let x, y be strings over Σ and L be a language over Σ . We say that x, y are distinguishable by L if $\exists z \in \Sigma^*$ such that $xz \in L$ and $yz \notin L$ or vice versa. ($xz \notin L$ and $yz \in L$)

If x, y are not distinguishable by L , we say that they are indistinguishable by L , and denote it by $x \equiv_L y$.



for any $z \in \Sigma^*$, we now have

$$\begin{aligned} \delta^*(q_0, xz) &= \delta^*(\delta^*(q_0, x), z) \\ &= \delta^*(r, z) \\ &= \delta^*(\delta^*(q_0, y), z) \\ &= \delta^*(q_0, yz) \end{aligned}$$

So $xz \in L \Leftrightarrow yz \in L$. So x, y are indistinguishable by L . So $x = y$. This is a contradiction.
So $\text{index}(L) \leq k$.

lemma 2: If $\text{index}(L) = k < \infty$, then there exists



So, what is lemma 1, lemma 1 says that if n is recognized by a DFA with k states then the index is at most k . So let us see, so what is index, index is the number of equivalence classes, another way to see it is the size of the largest pairwise distinguishable set. Suppose L has a DFA with k states, so, what we will do is that we will show that any two strings that end in the same state are indistinguishable.

Suppose there are k states. So, now we are saying that anything that ends in the same state, any strings that end in the same state are indistinguishable. So, suppose there are infinite strings that end in a certain state all of them are indistinguishable, meaning from the strings that end in that state I can only pick one string in the pairwise distinguishable set. So let us see. Suppose L is recognized by a DFA M with k states and suppose for the sake of contradiction the index is more than k .

Now we will show that we will show a contradiction to this assumption. This means that there is a set X that is of size more than k . That is what index means. There is a set X that is of size bigger than k which is pairwise distinguishable, meaning any two string that you take from X will be distinguishable by the language.

Now let q_0 be the starting state of the DFA. We assume there is a DFA and k is the number of states. Let us say k is 10 and suppose X has 11 strings. Now, for each of these strings in X you see where the DFA ends after reading them. There are 11 strings and there are only 10 states so at least two of these strings must end in the same state.

Same thing I am saying here, by pigeonhole principle there exist two distinct strings $x, y \in X$, distinct so x is not equal to y , such that they end at the same state, so $\delta^*(q_0, x) = \delta^*(q_0, y)$.

This of course will happen because there are more than k strings in capital X and there are only k states so at least two of them should kind of coincide as to where they end.

So now, x and y end in the same state. By assumption, x and y are distinguishable by the language but now we will contradict that by showing that they are actually indistinguishable by the language. As far as the DFA is concerned both of them ended in the same state, now whatever z you append to the to the x or to the y , we will show that wherever the DFA goes with when it sees xz , let us say where it goes for any $z \in \Sigma^*$. Where does the DFA end? That is exactly what is given by the notation here $\delta^*(q_0, xz)$. This is the state where xz ends up being.

So, another way to see this is you first see where x takes the DFA followed by where does z take it from there. So, suppose x takes it to, suppose the state r and y also takes it to r . Now from r , where does z take it to, but what is r ? r is also $\delta^*(q_0, y)$ and this is nothing but where the DFA ends up when it sees yz .

$$\begin{aligned} \delta^*(q_0, xz) &= \delta^*(\delta^*(q_0, x), z) \\ &= \delta^*(r, z) \\ &= \delta^*(\delta^*(q_0, y), z) \\ &= \delta^*(q_0, yz) \end{aligned}$$

So, what it means is that starting from the starting state, xz takes it to some state, let us say t and yz also goes to the same state t . Meaning, if t is an accepting state both xz and yz are accepted, if t is not an accepting state neither xz is accepted nor yz is accepted.

So, either both of them are in L or neither of them are in L , that is what I have written here. So, $xz \in L \Leftrightarrow yz \in L$. If xz is in L then yz is in L , if xz is not in L then yz is not in L . That is what I have written here, it is if and only if. So, which means that whatever z you put at the end of x and y it is not going to distinguish because as far as the DFA is concerned, where did x take the DFA to and where did y take the DFA to, that is all that matters.

Now from there it sees the same string z so it follows the same trajectory and ends in the same state. But we know that x and y took it to the same state r hence they are indistinguishable. This means x is, so not pairwise indistinguishable that is an error, so x and y are indistinguishable by L , that is what the definition was. I will just erase it and write not distinguishable, indistinguishable because whatever z you put it is not going to distinguish them.

So, this is a contradiction because capital X was supposed to be a pairwise distinguishable set but here, we have two distinct members of capital X which are indistinguishable. So, that is a contradiction. The assumption that it contradicts was that the index of L was strictly greater than k, hence the assumption is wrong and hence the index is less than or equal to k. So, we showed that if L has a DFA with k states the index is at most k which is the statement of lemma 1.

(Refer Slide Time: 11:14)

lemma 2: If $\text{index}(L) = k < \infty$, then there exists a DFA with k states that recognizes L.

Proof: Suppose $\text{index}(L) = k < \infty$. We will construct a DFA M with k states that recognizes L. let $X = \{x_1, x_2, \dots, x_k\} \subseteq \Sigma^*$ be a set of strings pairwise distinguishable by L.

$M = (Q, \Sigma, \delta, q_0, F)$. $Q = \{q_1, q_2, \dots, q_k\}$

Each state $q_i \in Q$ corresponds to $x_i \in X$.


Notation: $\delta(q, x)$ for $x \in \Sigma^*$ denotes the state reached by the DFA starting from q and reading the string x.


lemma 1: If L is recognized by a DFA with k states, then $\text{index}(L) \leq k$.


Proof: We will show that any two strings that end in the same state are indistinguishable.


Suppose L is recognized by a DFA M with k states. Suppose, for the sake of contradiction that $\text{index}(L) > k$.

This means there exists X such that X is pairwise distinguishable by L, and $|X| > k$.









a DFA with k states that recognizes L .

Proof: Suppose $\text{index}(L) = k < \infty$. We will construct a DFA M with k states that recognizes L . Let $X = \{x_1, x_2, \dots, x_k\} \subseteq \Sigma^*$ be a set of strings pairwise distinguishable by L .



$M = (Q, \Sigma, \delta, q_0, F)$. $Q = \{q_1, q_2, \dots, q_k\}$ $x_i a$

Each state $q_i \in Q$ corresponds to $x_i \in X$.

For each $a \in \Sigma$, $\delta(q_i, a)$ is defined as follows.

We have $x_i a \equiv_L x_j$ for some $x_j \in X$. Else, we can add $\{x_i a\} \cup X$ to get a larger pairwise distinguishable

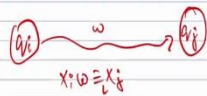
set. Now set $\delta(q_i, a) = q_j$.

For each $a \in \Sigma$, $\delta(q_i, a)$ is defined as follows.

We have $x_i a \equiv_L x_j$ for some $x_j \in X$. Else, we can add $\{x_i a\} \cup X$ to get a larger pairwise distinguishable set. Now set $\delta(q_i, a) = q_j$.

Similarly, $\epsilon \equiv_L x_m$ for some $x_m \in X$. Let $q_0 = q_m$.

Finally, define $F = \{q_i \mid x_i \in L\}$. Now we need to show that M recognizes L .



We first state a claim.

Claim: $\delta^*(q_i, w) = q_j \iff x_i w \equiv_L x_j$ for all i, j and $w \in \Sigma^*$

We will prove the lemma using the claim.

Suppose $x \in L$. Then $x \equiv_L x_i$ for some $x_i \in X \cap L$.

$x \equiv_L x_i \iff \delta^*(q_0, x) = q_i \in F$.

Therefore x is accepted by M .

Suppose $x \notin L$. Then $x \equiv_L x_i$ for some $x_i \in X$, and



We will prove the lemma using the claim.

Suppose $x \in L$. Then $x \equiv_L x_i$ for some $x_i \in X \cap L$.

$$x \in L \equiv_L x_i \Leftrightarrow \delta^*(q_0, x) = q_i \in F.$$

Therefore x is accepted by M .

Suppose $x \notin L$. Then $x \equiv_L x_i$ for some $x_i \in X$, and $x_i \notin L$. Similar to the above, we get that

$$\delta^*(q_0, x) = q_j, \text{ where } q_j \notin F. \text{ So } M \text{ does not accept } x.$$

Thus M recognizes the language L .



The next statement or next lemma that we have to show is lemma 2 which is opposite. In lemma 1 we said that if there is a DFA that recognizes L with k states, then the index is at most k . In lemma 2 we say the opposite. If index is at most k or index is equal to k then there is a DFA with k states that recognizes L . So, if the index is 10 then we can construct a DFA with 10 states.

So, what we do here is we actually construct a DFA of using the index. So, suppose the index is k which is a finite number, now we are going to construct a DFA. So let $X = \{x_1, x_2, \dots, x_k\}$ be a set of pairwise distinguishable strings. So, this is the largest set of pairwise distinguishable strings, so that is the definition of index. Index is the size of the largest pairwise distinguishable set X by L . Index is k so we can make a set of size k which is $\{x_1, x_2, \dots, x_k\}$.

So now what we will do is we will use this set X to construct the DFA. Let us construct the DFA. So let $M = (Q, \Sigma, \delta, q_0, F)$ be the DFA. So, we already know what Σ is and Q is the set of states and the set of states is going to be $\{q_1, q_2, \dots, q_k\}$. There are going to be k states, which is what we want to show. If the index is k , there is a DFA with k states.

Now q_1 to q_k , each of these states, so q_1 will correspond to x_1 , q_2 will correspond to x_2 and so on q_k corresponds to x_k , so each q_i corresponds to the respective x_i , let us see how in a moment. So now the next thing we need to define is how the arrows are defined. So let us take q_i and then it sees a symbol a . Where do you end up? This is the question. So, this transition needs to be defined. Notice this that q_i corresponds to x_i so let us take x_i .

Now let us append a to x_i . So x_i is some string. When you append it, you get a bigger string. Now this string $x_i a$ which I am underlining over here, it is equivalent to some other string in

capital X. So capital X is the largest set of strings that are pairwise distinguishable so $x_i a$ will be equivalent to some string in that set X because if it is not equivalent to some string in that set then you can add $x_i a$ to the set and get a bigger pairwise distinguishable set but then that is not possible because we assumed or by definition this is the biggest pairwise distinguishable set.

So, $x_i a$ must be equivalent to some string in capital X. Let that string be x_j . Now $x_i a$ is equivalent to or is indistinguishable with some string in capital X. Now that string, let us say it is x_j . Now, what we do is we ask the question where does this arrow go to from q_i upon reading, this is symbol a. The answer is it goes to q_j . How did we get q_j ? We saw what is equivalent to x_i followed by a that is how we get the transitions.

So, now for all the states q_1 to q_k and for all the symbols in the alphabet a, b, c whatever 0, 1, whatever be the alphabet we do this to find out where the arrow points to. So that gives you the transition so I have defined the states q_i , I have defined the transitions δ , Σ is already known. Now what remains to be shown is define this starting state q_0 and the accepting states F.

Just like I said before, the empty string ϵ is equivalent or is indistinguishable from some string in the set X. The empty string epsilon is equivalent to some string in the set X, let us call that string x_m . Why does it have to be equivalent? Because if it is not equivalent it is distinguishable from all the other strings in X, you can add the empty string to X and get a bigger pairwise distinguishable set which is not possible.

So empty string better be equivalent to some string already there, equivalent meaning indistinguishable with some string that is already there in X and suppose it is x_m . Suppose the empty string is equivalent to x_m then the corresponding state q_m is to be set at the starting state. So empty string will be equivalent to some state, some string x_m and the corresponding state is the starting state of the DFA. Finally, the last thing needs to be defined. We have defined a set of states, the alphabet is already known, transition is known, starting state is defined. The only thing left is the accepting states. Accepting states is fairly straightforward.

We have x_1 to x_k strings and correspondingly we have q_1 to q_k states. Some of these x_1 to x_k are in the language L. Suppose x_1 is in L. If x_1 is in L then you make q_1 accepting state. If x_2 is not in L you make q_2 not an accepting state. If x_3 is not in L you make q_3 not an accepting state. If x_k is in L you make q_k an accepting state.

So, whichever x_i is in the language that q_i will be an accepting state. Whichever x_i is not in the language that q_i is not in the accepting states. Now all that we need to do, so we have defined the DFA we have told what is Q , what is δ , what is Σ , what is q_0 , what is F . All that we need to do is to show that this DFA recognizes our language.

We will first state this following claim which is kind of what we are building towards. The claim is that suppose upon reading the string w from the state q_i you move to q_j . So $\delta^*(q_i, w) = q_j$. When you are in q_i and then you read w and then you get to q_j . I am reading squiggly lines because it may not be one transition. It could be multiple transitions,

$$\delta^*(q_i, w) = q_j \Leftrightarrow x_i w \equiv_L x_j \text{ for all } i, j \text{ and } w \in \Sigma^*$$

This means that if this happens, this happens if and only if $x_i w$ is equivalent to x_j . So, these two things are equivalent or these two things happen together or do not happen together. If one of them happens, the other one also happens. Let us see what do we want to do now. We assume that by the assumption the index is k . We wanted to show that there is a DFA with k states, we actually constructed a DFA with k states. We want to show that this DFA recognizes the language L .

So, what we will do now is to assume that this claim is true. Okay, we will assume that this claim is true and then we will prove that this DFA recognizes L . After which, we will prove the claim. The claim is fairly straightforward, fairly inductive, standard inductive proof. So let us just forget the proof of the claim for a moment and we will prove that the DFA constructed recognizes the language. Let us see how.

So, how do we show that the DFA recognizes language? We will show that every string in the language is accepted and every string that is not in the language is not accepted. Suppose x is in the language. Then it follows that it is equivalent to some string that is in the language, some string that is in the set X .

Because x is equivalent to some string in the language and some string in the set X , because X is the largest pairwise distinguishable set and whatever it is equivalent with will have to be a string in the language. So, x is equal to some x_i that is both in X as well as in the language. Which means that x can be viewed as empty string followed by x and we know that this is equivalent to x_i .

$$x = \epsilon x \equiv_L x_i$$

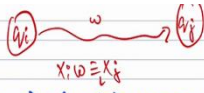
So ϵx is equal to x_i . By the claim that we just said, this means that empty string followed by x is equal to x_i , this means that $\delta^*(q_0, x) \equiv_L q_i \in F$ because q_0 is a state corresponding to ϵ the empty string that is how we constructed, x is the same, q_i is the state corresponding to x_i . So, $\delta^*(q_0, x) \equiv_L q_i$ meaning from the starting state upon reading the symbol x or upon reading the string x the machine ends at or the DFA ends at q_i . And by the definition of the accepting states F .

So, what were the accepting states? All the x_i 's that were in the language correspond to all the q_i that are accepting states. So, since by assumption x_i is a string in the language, q_i is in the language. Therefore, upon reading the string x the DFA ends at q_i , which is an accepting state. So, x is accepted. The next part is fairly similar. We have to show that any string that is not in the language is rejected. Suppose $x \notin L$. This means $x \equiv_L x_j$ that is in X . This x_j must be in X but it is not a member of the language because otherwise they will not be equivalent.

Similar to above we can do the exact same thing, x is equal to ϵx etcetera. We will get using the claim that $\delta^*(q_0, x) \equiv_L q_j$ meaning upon reading the string x from the starting state the DFA ends at q_j . Like before, since x_j is not in the language, q_j is not an accepting state. q_j is not an accepting state which means the string X takes the DFA to a state q_j which is not an accepting state and hence it means that the DFA does not accept the string x .

We considered an x that is not in the language, we showed that M is not accepting. M does not accept this string. So whenever x is in the language M accepts whenever x is not in the language M does not accept. This means that M recognizes the language. So fairly straightforward proofs once we assume the claim. The claim is also sort of technical so let me just recap the lemma very quickly, before moving to the proof of the claim.

The lemma statement is that if the index is equal to k there is a DFA with k states that recognizes the language. The proof is constructive. We actually construct a DFA with k states. So, the index is k means that there is a set of k pairwise distinguishable strings. We make a DFA where each state corresponds to each of these k strings and we define the transitions, accept state, start state, everything correspondingly. Start state is the state that corresponds to the string equivalent to the empty string, accept states are the states that correspond to the strings that are in the language. And then we use that to show that any strings in the language L are accepted, any strings that are not in the language are not accepted. (Refer Slide Time: 25:55)



Claim: $\delta^*(q_i, w) = q_j \iff x_i w \equiv_L x_j$ for all i, j and $w \in \Sigma^*$.

Proof: By induction on $|w|$.

When $|w|=0$, $w = \epsilon$.

$\delta^*(q_i, \epsilon) = q_i$ and $x_i \epsilon = x_i \equiv_L x_i$

When $|w|=1$, $w = a \in \Sigma$.

Let $\delta^*(q_i, a) = \delta(q_i, a) = q_j$

By definition of δ , we have $x_i a \equiv_L x_j$.

$X = \{x_1, x_2, \dots, x_k\} \subseteq \Sigma^*$ be a set of strings pairwise distinguishable by L .



$M = (Q, \Sigma, \delta, q_0, F)$. $Q = \{q_1, q_2, \dots, q_k\}$ $x_i a$

Each state $q_i \in Q$ corresponds to $x_i \in X$.

For each $a \in \Sigma$, $\delta(q_i, a)$ is defined as follows.

We have $x_i a \equiv_L x_j$ for some $x_j \in X$. Else, we can add $\{x_i a\} \cup X$ to get a bigger pairwise distinguishable set. Now set $\delta(q_i, a) = q_j$.

Similarly, $\epsilon \equiv_L x_m$ for some $x_m \in X$. Set $q_0 = q_m$.

Claim: $\delta^*(q_i, w) = q_j \iff x_i w \equiv_L x_j$ for all i, j and $w \in \Sigma^*$.

Proof: By induction on $|w|$.

When $|w|=0$, $w = \epsilon$.

$\delta^*(q_i, \epsilon) = q_i$ and $x_i \epsilon = x_i \equiv_L x_i$

When $|w|=1$, $w = a \in \Sigma$.

Let $\delta^*(q_i, a) = \delta(q_i, a) = q_j$

By definition of δ , we have $x_i a \equiv_L x_j$.

So claim is true.





Induction step. let $|w| = l > 1$. let $w = va$ where
 $|w| = |v| + 1$ and $a \in \Sigma$.
 $\delta^*(q_i, w) = \delta(\delta^*(q_i, v), a) = \delta(q_{j_1}, a) = q_{j_2}$
 where $q_{j_1} = \delta^*(q_i, v)$.
 By induction, we have $x_{j_1} \equiv_L x_i v$, and
 by definition of δ , $x_{j_2} \equiv_L x_{j_1} a$
 $x_{j_2} \equiv_L x_{j_1} a \equiv_L x_i v a = x_i w$.
 So claim holds for w , $|w| > 1$, as well.



And what remains to be shown is the following claim. The claim is something like this. So, suppose the string w takes the DFA from q_i to q_j then $x_i w$ is equivalent to x_j and vice versa. If $x_i w$ is equivalent to x_j then w takes the string from q_i to q_j and this is fairly straightforward this is done by induction on the length of w .

The base cases are simple so I will do two base cases 0 and 1. The induction is on the length of w so the first base case is when the length of w is 0. There is only one string that is of zero length which is the empty string ϵ . So, what is $\delta^*(q_i, \epsilon)$, meaning where does empty string take the DFA from q_i ? It takes it to itself and trivially $x_i \epsilon \equiv_L x_i$. This is what we have to show, if ϵ takes it to some other state then x_i also is equivalent to the corresponding state.

So ϵ takes it to q_i because ϵ cannot move the states in a DFA. But then x_i followed by ϵ is equivalent to x_i itself, so hence this is true in the case of w equal to ϵ . When $|w|$ is equal to 1 which means the length of the string is 1, which means w is a symbol from the alphabet. w is some symbol a from the alphabet. Now $\delta^*(q_i, a)$, meaning where does a take the DFA starting from q_i . This is just a transition function because it is a single symbol, not a string. So, it is the same as $\delta(q_i, a)$, because we can remove the star.

$$\delta(q_i, a) \equiv_L q_j$$

If you remember the way we define $\delta(q_i, a) \equiv_L q_j$ was by checking what is the equivalence of $x_i a$. What is $x_i a$ equivalent to? We set it to be q_j because $x_i a$ was equivalent to x_j . So, by the definition of the transition function $x_i a \equiv_L x_j$, so the claim is true here as well. So, this is

just by the definition of the transition function. Now, the main thing is the induction. So, we showed it for the length $|w| = 0$ and $|w| = 1$.

Now the next thing is the induction step. Suppose $|w| = l > 1$ and let us say w is of the form $v a$ where v is a string and a is a symbol. So $|w| = |v| + 1$. So v could be some string. The length $|v| = l - 1$ and a is a single symbol from the alphabet. So, where does w take the DFA starting from q_i . First, we will see where does v take the DFA and then we will see where does a take the DFA.

We can break down w as v and followed by a . Suppose v takes the DFA to q_{j_1} and then a takes it to q_{j_2} . So, q_{j_1} , as I said before is where v takes the DFA to and q_{j_2} is where a takes it from q_{j_1} . By induction, $q_{j_1} = \delta^*(q_i, v)$. By induction it follows that $x_{j_1} \equiv_L x_i v$ and by definition of δ we have $x_{j_2} \equiv_L x_{j_1} a$. Because, by the definition of δ , we have $\delta(q_{j_1}, a) = q_{j_2}$.

So now x_{j_2} is equivalent to $x_{j_1} a$. This follows by what I have written here, the first equivalence. The second equivalence is that $x_{j_1} \equiv_L x_i v$. Maybe I will mark it with green colour. This is the green equivalence, x_{j_1} is equivalent to $x_i v$. But then, $x_i v a$ is simply $x_i w$ as it is and $v a$ is the same as w . Maybe, I will use another colour, this red colour. This is equal to $x_i w$, not equivalence.

$$x_{j_2} \equiv_L x_{j_1} a \equiv_L x_i v a = x_i w$$

So now we have shown that x_{j_2} is equivalent to $x_i w$ where q_{j_2} was a state reached from q_i , starting at q_i upon reading the string w , which is what we wanted to show. We wanted to show that if the DFA goes to q_j starting from q_i by reading the string w then $x_i w$ is equivalent to x_j . Here, the state that it went to was q_{j_2} and here we are saying that $x_i w$ is equivalent to x_{j_2} . So, this means the claim holds for the inductive step as well and that completes the proof.

So, if you find the induction too technical my suggestion is to probably forget the induction for a moment and just try to understand the rest of the proof. The rest of the proof is fairly nice and easy to follow. Try to just believe the claim. Forget about worrying about the proof of the claim. Try to believe this claim that is in the red box and try to see how the construction of the DFA works. The construction mainly works by, for each string that is in the pairwise distinguishable set capital X we make a state and we make the transitions accordingly and that somehow magically works together. It is not magical but it is somewhat sensible as well.

So, try to follow this proof without worrying about the proof of this claim and once you understand that proof then you can come to understanding the proof of the claim. So, the proof of the claim is also fairly standard but try to separate the two so that if together it becomes too hard to follow, that is one suggestion. What we showed is the proof of lemma 1 and lemma 2.

(Refer Slide Time: 33:34)

reached by the DFA starting from q_0 and reading the string x .

lemma 1: If L is recognized by a DFA with k states, then $\text{index}(L) \leq k$.

Proof: We will show that any two strings that end in the same state are indistinguishable.

Suppose L is recognized by a DFA M with k states. Suppose, for the sake of contradiction that $\text{index}(L) > k$.

This means there exists X such that X is pairwise distinguishable by L , and $|X| > k$.

Let q_0, q_1, \dots, q_k be the states of M . Then, q_0, q_1, \dots, q_k are states, then $\text{index}(L) \leq k$.

Proof: We will show that any two strings that end in the same state are indistinguishable.

Suppose L is recognized by a DFA M with k states. Suppose, for the sake of contradiction that $\text{index}(L) > k$.

This means there exists X such that X is pairwise distinguishable by L , and $|X| > k$.

Let q_0 be the starting state of M . By pigeonhole principle, there exists two strings $x, y \in X$, $x \neq y$ such that $\delta^*(q_0, x) = \delta^*(q_0, y)$ (x and y end in the same state).



Let q_0 be the starting state of M . By pigeonhole principle, there exists two strings $x, y \in X$, $x \neq y$ such that $\delta^*(q_0, x) = \delta^*(q_0, y)$ (x and y end in the same state)

For any $z \in \Sigma^*$, we now have


$$\begin{aligned}\delta^*(q_0, xz) &= \delta^*(\delta^*(q_0, x), z) \\ &= \delta^*(r, z) \\ &= \delta^*(\delta^*(q_0, y), z) \\ &= \delta^*(q_0, yz)\end{aligned}$$

So $xz \in L \Leftrightarrow yz \in L$. So x, y are indistinguishable by L . So $x \in \underline{y}$. This is a contradiction.
So $\text{index}(L) \leq k$.




So, lemma 1 said that if L is recognized by a DFA with k states then the index is at most k , this was shown by assuming that this is not the case and then by showing a contradiction. If the index was greater than k then we showed that two strings from the pairwise distinguishable set must end at the same state. Then we said that anything that ends at the same state must be pairwise indistinguishable which contradicts what we said earlier.

(Refer Slide Time: 34:10)




Lemma 2: If $\text{index}(L) = k < \infty$, then there exists a DFA with k states that recognizes L .

Proof: Suppose $\text{index}(L) = k < \infty$. We will construct a DFA M with k states that recognizes L . Let $X = \{x_1, x_2, \dots, x_k\} \subseteq \Sigma^*$ be a set of strings pairwise distinguishable by L .




$M = (Q, \Sigma, \delta, q_0, F)$. $Q = \{q_1, q_2, \dots, q_k\}$



Then we said lemma 2. Lemma 2 said that if the index is k then there is a DFA with k states which we proved in a constructive fashion and that together imply the Myhill-Nerode theorem.

(Refer Slide Time: 34:23)



Example: $A = \{0^n 1^m \mid n \geq 0\} = \{\epsilon, 01, 0011, 000111, \dots\}$

Consider $x_i = 0^i$ for $i = 0, 1, 2, 3, \dots$


The set $X = \{x_i \mid i \geq 0\}$ is pairwise distinguishable by A .

$$X = \{0^i \mid i \geq 0\} = \{\epsilon, 0, 00, 000, \dots\}$$

Given x_i, x_j such that $i \neq j$.

Consider the string 1^i . We have $x_i 1^i \in A$, but $x_j 1^i \notin A$ when $i \neq j$. So 1^i distinguishes x_i and x_j . So X is an infinite set, pairwise distinguishable by A .

Hence A is not regular.



hence n is not regular.

$$A = \{0^n 1^n \mid n \geq 0\}$$

$$X = \{\epsilon, 0, 00, 000, 0000\} \rightarrow$$

$01 \in A$ and $001 \notin A$. So 0 and 00 are distinguishable by A .

$0011 \in A$ but $00011 \notin A$. So 00 and 000 are distinguishable by A .

We can verify that above X is pairwise distinguishable by A .



Myhill-Nerode Theorem

(John Myhill and Anil Nerode, 1958)

Not in the book. Only stated as exercise 1.51 and 1.92.

This provides a necessary and sufficient condition for a language to be regular.

Def 1: Let x, y be strings over Σ and L be a language over Σ . We say that x, y are distinguishable by L if $\exists z \in \Sigma^*$ such that $xz \in L$ and $yz \notin L$ or vice versa. ($xz \notin L$ and $yz \in L$)



Finally, just a brief example. In fact, part of this example we touched upon in the lecture 14. So, consider the language is $A = \{0^n 1^n \mid n \geq 0\}$. This is a language. So, this as we saw in lecture 13 this is $\{\epsilon, 01, 0011, \dots\}$, and we know this is not a regular language. Let us see why Myhill-Nerode theorem implies that this is not a regular language.

So basically, to show that this is not a regular language, we need to show that the index is not finite. So, consider this following set $X = \{x_i \mid i \geq 0\}$ where $x_i = 0^i$. X is basically 0^i for all i . So, it is just consisting of strings $\{\epsilon, 0, 00, 000, \dots\}$. This is an infinite set. The claim is that this is a pairwise distinguishable set.

So, why is it a pairwise distinguishable set? We kind of saw the proof already in lecture 14. Consider these four strings $0, 00, 000, 0000$, so for anything, if you try to append 1, 01 is in the

language but 001 is not in the language, 0001 is not in the language and 00001 is also not in the language. So, the first string 0 is distinguishable from any other string in the set.

And similarly, if you consider appending 11, so 011 is not in the language but 0011 is in the language, hence the second string 00 is pairwise distinguishable from all the other strings. It is the same argument again. If you see this set X here it is an infinite set it contains $\epsilon, 0, 00, 000, \dots$. The string 1^i will ensure that just x_i alone is in the language. $x_i 1^i$ will be in the language but $x_j 1^i$ will not be in the language whenever $i \neq j$. $x_j 1^i$ is not in the language. Hence, 1^i distinguishes x_i and x_j .

So, if you take any x_i and x_j the 1^i will distinguish them. Hence all the strings in the set capital X are pairwise distinguishable. Hence this is an infinite set that is pairwise distinguishable which means the index of the language A is infinite.

And that is kind of is consistent with what we know, which is that A is not a regular language. The index is infinite, hence A is not regular. This is something that we have already said in lecture 14. Anyway, this concludes the part on Myhill-Nerode theorem which is a necessary and sufficient condition to show that a language is regular. This is a necessary and sufficient condition. We defined this equivalence relation and the condition is that A is regular if and only if the equivalence relation defined by A results in a finite number of equivalence classes.

And with that, we come to the end of the part on regular languages. This is the last topic in the first chapter which is a chapter on regular languages. In fact, this part, this Myhill-Nerode theorem is in fact not there in the textbook per se. It is listed as an exercise but this completes the part on regular languages. So, the next thing that we will see are context-free languages which are slightly more sophisticated or more intricate or more involved than regular languages.

So regular languages are fairly simple, regular expressions and DFAs and NFAs. Context-free languages involve another dimension. There are more things that you can do with context-free languages. So, we will see that in the next lecture.