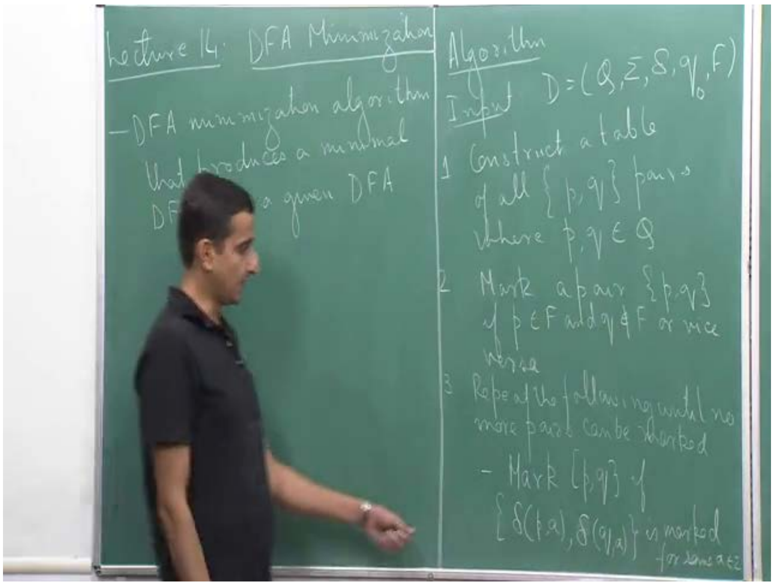
Theory of Computation
Prof. Raghunath Tewari
Department of Computer Science and Engineering
Indian Institute of Technology, Kanpur

Lecture -14
DFA minimization

Welcome to Lecture 14 of this course. So, today we are going to talk about DFA minimization. So, given a regular language, and given DFA for the regular language how can we figure out whether it is the smallest DFA that is accepting the language. Also if it is not the smallest, how can we construct the smallest DFA that accepts the same language?

In other words, given one DFA, how can I construct another DFA that is that are equivalent in the sense that both of them accept the same language and the second DFA has the minimal number of states. This is what we are going to see today. And what we will show is a constructive way of showing this that whether DFA is the minimum DFA or not, in the sense that we will give an algorithm that takes the DFA and outputs a minimal DFA out of it.

(Refer Slide Time: 01:23)

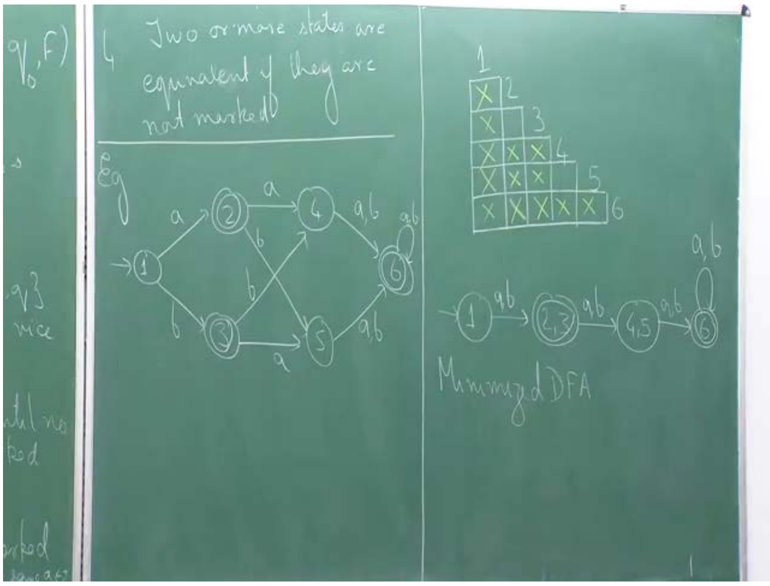


We give this DFA minimization algorithm that produces a minimal DFA from a given DFA. So, how do we do this? What I am going to do is first I will give the algorithm, I will describe what the algorithm is and then we will look at a couple of examples to get a better understanding as to how it works. Let us look at the algorithm. First construct, so let say that your input is a DFA; D equal to Q, sigma, delta, q naught and F, this is the input that is given to you. Now what we do is first we construct a table of all p comma q pairs, where p and q are states in the DFA.

So construct a table which has all pairs of form p comma q and these are unordered pairs. So, p comma q and q comma p is the same object, so that is why I wrote it in set theoretic notation. So, look at all those pairs where these elements are states of my DFA. Now for so mark a pair p comma q, if p is a accept state, and q is not an accept state or vice versa. So, look at all these pairs and then mark a pair p comma q, if one of them is an accept state and other is not an accept state. So, may be p accept, q naught not accept; or p is not an accept state and q is an accept state; we mark all those pairs. Now the third step is repeat the following until no more pairs can be marked.

So, we repeat the following for sub routine what do we repeat. So, mark p comma q, if delta p comma a comma delta q comma a is marked for some a in sigma. So, mark all pairs such that if you look at p comma q then if you look at unmarked pairs p comma q, then the pair delta p comma a and delta q comma for some a belonging to sigma has already been marked before. So, if you go over all the pairs, all the unmarked pairs and you search whether you can mark a new pair in this manner, and you keep on doing this. So, you keep on repeating this process until no new pair can be marked.

(Refer Slide Time: 07:05)



At the end, two states p and q are equivalent if they are not marked ok. So, look at all the unmarked pairs in this table, and you say that two states are equivalent; it can be more than two also. So, two or more so maybe I should say here that two or more states are equivalent. I can have p q r, p q r. So, two or more states are equivalent, if finally, they are unmarked. So, what we mean by this let us look at an example, it will be clearer.

Let us look at a DFA. I will construct the following DFA. We have 1 as the start state, and then from 1 on an a we go to 2; on a b, we go to 3; from 2 on a, we go to 4; and on a b, we go to 5, from 3 on a, we go to 5 and on a b, we go to 4. And from 4, both on a and b, we go to 6; and again from 5 both on a and b, we go to 6. And at 6, we just stay on itself on a and b. So, from 6, if you see an and b, we stay at itself. And they accept states are 2, 3 and 6. Here, we have a DFA, how can we minimize the DFA.

First let us construct the table. We will write 1, 2, 3, 4, 5 and 6. Here, we have the table. Now, let us mark the pairs. So, in the first iteration, if you look at step 2 of the algorithm, we mark all pairs p comma q such that one is an accept state and the other is not an accept state. So, we have our accept states as 2, 3, 6, 2 3 and 6. I mark 1, 2; I mark 1, 3 and I mark 1, 6. Again between 2 and 3, I do not mark 2 comma 3 because both are accept states, but I mark 2, 4 and I mark 2, 5. Again I do not mark 2, 6 because both are accept states. Between 3 and 4, and I mark 3, 4; it is an accept state; I mark 3, 5 it is an accept state, I do not mark 3, 6. I will mark 4 comma 6, which is an accept state a one is accept and other is not accept and I will mark 5 comma 6. So, this is round one.

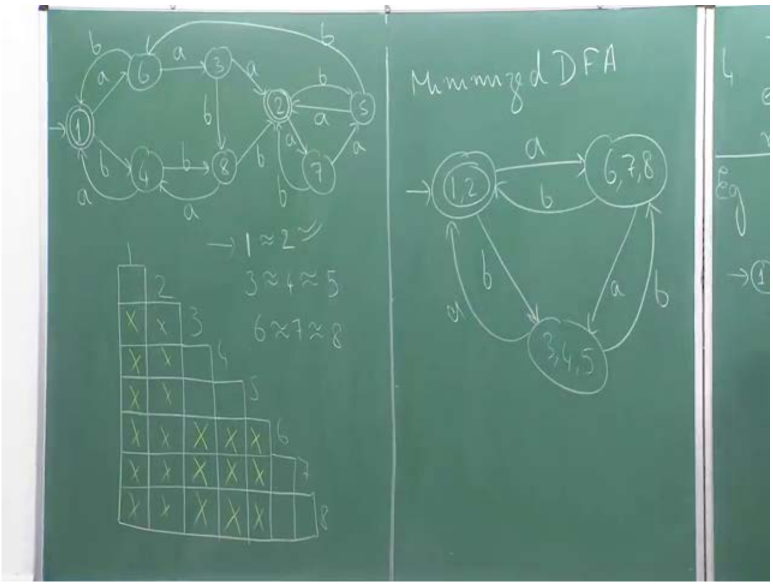
In the second round, again we will go over all the unmarked pairs. Let us look at the first unmarked pair 1 comma 4. So, from 1 comma 4 on an a, where we go to. So, from 1 comma 4 on an a, we go to 2 comma 6. So, 2 comma 6 is not marked. We cannot do anything. Again on 1 comma 4 on b, we go to 3 comma 6. Once again 3 comma 6 is not marked. So, we cannot mark 1 comma 4. Let us look at the next unmarked pair 1 comma 5. So, from 1 comma 5 on a, we go to 2 comma 6, which is not marked and from 1 comma 5 on a b, we go to 3 comma 6, again that is not marked. So, we cannot mark 1 comma 5.

Now let us go to 2 comma 3. So, from 2 comma 3 on an a, we go to 4 comma 5, which is not marked; and on a b again, we go to 4 comma 5 which is not marked. So, we cannot marked 2 comma 3. The next is 2 comma 6. Let us look at 2 comma 6. So, from 2 comma 6 on an a, we go to 4 comma 6, but now if you see 4 comma 6, 4 comma 6 is already marked. So, therefore, we mark 2 comma 6 in the second round. Let us look at 3 comma 6. So, again from 3 comma 6 on an, I go to 5 comma 6, which is already marked hence I mark 3 comma 6. Now, we have 4 comma 5. So, from 4 comma 5 on an a, I go to 6 itself, which by a definition is 6 comma 6 let say, and 6 comma 6 by definition is not marked. So, we do not mark.

And again from 4 comma 5 on a b, I go to 6 comma 6 which is not marked, so I do not mark 4 comma 5. Because in this iteration, there was at least one pair which got marked, in fact, there were 2 pairs which got marked. Therefore, I will again repeat my this marking algorithm. Again I look at 1 comma 4. So, from 1 comma 4, I go to 2 comma 6 on an a, but now observe that 2 comma 6 has been marked. Therefore, 1 comma 4 also gets marked. Again from 1 comma 5 on an a, I go to 2 comma 6, which is marked. So, I mark 1 comma 5. And again you can check that from 2 comma 3 both on a and b, I will go to states or I will go to pairs which are not marked. And the same things for 4 comma 5; from 4 comma 5, I will go to pairs which are not marked. So, this will not get marked.

Again if I repeat the iteration, these two will remain unmark and therefore the algorithm terminates over here. Now according to the fourth step, if two or more states are equivalent they are unmarked. In this case, 2, 3 are unmarked and 4, 5 are unmarked. So, the new DFA will be or the minimized DFA will be I will have a state for 1, which is my start state. I will have a single state I will call it 2 comma 3. So, from 1 I go to 2 comma 3 both on a and b. Again 4 and 5 are equivalent, so I have write has 4 comma 5. From 2, 3, I go to 4 comma 5 on a b. And from 4 comma 5, I go to 6 again on an a b; and I stay there on a b. What are my accept state, the accept states are the old accept states. So, 6 is an accept state, and because 2 comma 3 was an accept state earlier, here it has been collapsed into a single state. So, 2 comma 3 is also an accept state. This is how you construct the minimized DFA.

(Refer Slide Time: 17:19)



Let us look at one more example. So, I will have a DFA. I will have 1 which is a start state and from 1 an a, I go to 6; on a b, I go to 4. From 6 on an a, I go to 3; and on b, I come back to 1. From 4 on a, I go to 1; and on a b, I go to 8. Now from 3 on an a, I will go to a state 2; and on b, I will go to 8. From 8 on an a, I go to 4; it was b; and on b, I go to 2. So, it is done from 2 on a, I go to 7; and on a b, I will go to 5. From 7 on a, I go to 5; and on a b, I go back to 2. From 5 on a, we go to 2; and on b, we go to 6. So, this is the entire DFA and the accept states are 1 and 2. So, 1 is an accept state, and 2 is an accept state. This is the slightly more complicated example, it has 8 states, but the algorithm is the same.

Let us try to construct a table and we will mark the pairs of the table according to our algorithm. So, we have 1, 2, 3, 4, 5, 6, 7 and 8. So, this is the table that we have. And now let us mark the pairs. So, first I mark all pairs such that one is accept, and the other is not accepted. So, except for 1, 2 everything else will get marked, so 1 3, 1 4, 5, 1 6, 1 7 and 1, 8. Then again 2 is an accept state, so 2 3, 2 4, 2 5, 2 6, 2 7 and 2 8 gets marked and nothing else get marked in the first round. Now, we will star iterating.

So from 1 to observe that on an a, and both on an a and b, I go to an unmarked pair. So, 1, 2 does not get marked. From 3, 4, again the same thing happens; so on an a, I go to 1, 2; and on b, I will go to 8, 8 both are unmarked. From 3, 5 again the same thing. So from 3 5 on an a, I go to 2, 2; and on a b, I will go to 6, 8 both are unmarked, so 3 5. From 3, 6, if I take 3 and 6 then on an a, I go to 1 2, which is unmarked; and on a b, I will go to... So, from 3, 6 on an a, I go to 2 3; right 2 3 is marked hence 3, 6 gets marked. Let us look at this again. So, from 3, 6 on an a I go to again 3 comma 2, which is your marked state. So, this 1 gets marked.

Now let us look at 3, 7. So, from 3, 7 on a a, I go to 2, 5, which is marked. So, 3, 7 will get marked. From 3, 8 on a a, I go to 2, 4 which is marked. So, 3, 8 gets marked. From 4, 5 observe that from 4, 5 both on a and b, I do not go to mark states. So, 4, 5 will not get marked. Now let us look at 4, 6. So, from 4, 6 on a a, I go to 1, 3. So, 1, 3 is already marked. So, 4 6 gets marked. Again for 4, 7 on a a, I go to 1, 5 which is marked, of course, I am sorry. Let us I just made a small error. So, 4, 5 is not marked. So, 4, 6 is marked; and now 4, 7 gets marked. From 4, 8 on a a, I go to 1, 4 which is marked. So, 4, 8 gets marked.

Now, let us look at 5, 6. Now, let us look at 5, 6. So from 5, 6 on a a, we go to 3, 2 which is marked. So, 5, 6 gets marked. From 5, 7 on a a, we go to 5 and 2, which is marked, hence this gets marked. And also you check that 5, 8 will get marked. Now let us go to 6, 7. So, from 6, 7 on a a, we go to 3, 5 which is not marked; on a b, we go to from 6, 7, on a b, we go to 1, 2 which is also not marked. So, 6, 7 do not get marked. And you can just check here to see that 6, 8 will also not get marked and 7, 8 will also not get marked. So, there were some states which got marked. So, we again repeat the process. And again after repeating, you will see that 1, 2 will not get marked; 3, 4 will not get marked; 3, 5 will not get marked; 4, 5 will not get marked; and 6, 7, 6, 8, and 7, 8 will not get marked. So, this is the final state of the table.

Now we look at what are the equivalent states. Here, 1, 2 are not marked. So, 1 and 2 are equivalent. So, 1 is equivalent to 2. We have 3, 4 not marked which means that 3 is equivalent to 4. And also we have 3, 5 not marked which means that 3 is equivalent to 5 as well. So, by equivalence relation, it can be proved that this is actually an equivalence relation, so that need some work. I will not go into that part, but you can prove it. So, because 3, 4 is equivalent and 3, 5 are also equivalent, it means that 4 5 is equivalent which actually follows from the table 4, 5 is not marked.

So, 3, 4 and 5 are all equivalent which means that there will be one single state in my minimized DFA. And the same thing happens for 6, 7, 8. The 6 is equivalent to 7 which equivalent to 8 which means that this will form one single state in my minimized DFA. Finally, the minimized DFA will contain three states, one state for this 1, 2 pair; one state for 3 4 5 pair and one state for 6 7 8 pair, because 1 was the start state, this will be the start, and because again 1, 2 are accept states this will also be the accept state.

Let me draw the DFA. The minimized DFA will look like this. So, we have 1, 2 which is my start and accept state; on a a, I go to 6, 7, 8; and on a b, I go to 3, 4, 5. From 6, 7, 8 on a a, I go to 3, 4, 5; and on a b, I go back to 1 comma 2. And from 3, 4, 5 on a a, I go to 1 comma 2; and on a b, I go to 6, 7, 8. So, this is the DFA.

Thank you.