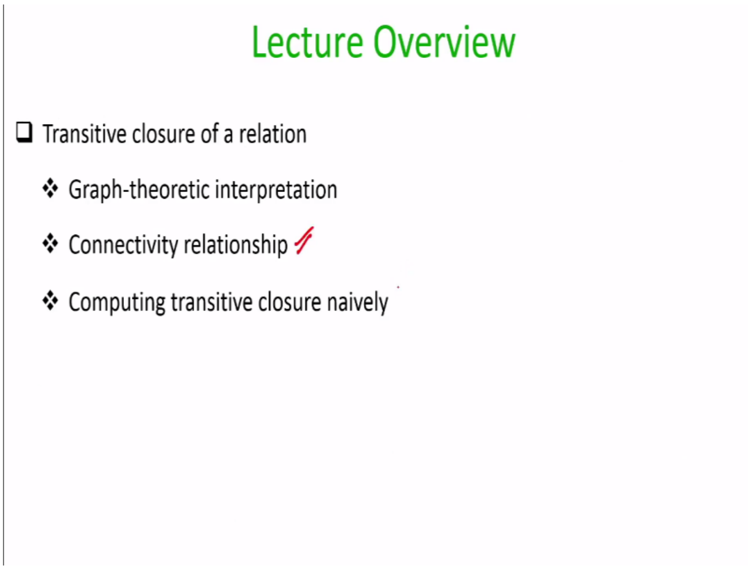
Discrete Mathematics
Prof. Ashish Choudhury
Department of Mathematics and Statistics
International Institute of Information Technology, Bangalore

Lecture -18
Transitive Closure of Relations

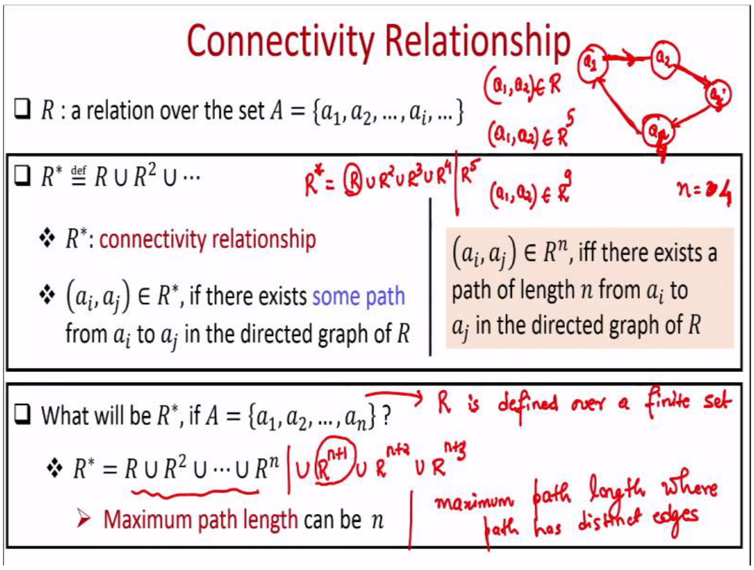
Hello everyone, welcome to this lecture. In this lecture, we will continue our discussion regarding how to construct a transitive closure of relations.

(Refer Slide Time: 00:27)



And for that, we will see some graph theoretic interpretation of transitive closures. We will discuss what we call as the connectivity relationship in the graph of a relation and then we will see a naive algorithm for computing the transitive closure.

(Refer Slide Time: 00:46)



So, let me start by introducing the connectivity relation. So, imagine you are given a relation R over a set which may or may not be finite. So, your R is a subset of A x A. And I define this relation R*, which I call as the connectivity relation and this is basically defined to be the union of different powers of your relation R. It turns out that, as per my definition of R* an element, a_i will be related to a_j in this relation R* provided there exists some path of any length of it may be of length 1, it may be of length 2, it may be of length 3.

I do not care about the length. The guarantee is that there exists at least one path from the node a_i to the node a_j in the directed graph of your relation R. And why so, because recall in the last lecture. We proved this statement by induction. The statement states that if you have element a_i related to element a_j in the nth power of your relation R.

Then that is possible only if you have a path of length n, from the node a_i to the node a_j in the directed graph of your relation R. Now, if I say that a_i is present in R* then it means that either (a_i, a_j) is present in R^1 or it is present be in R^2, and in the same way it will be present in some power of R. I do not know which power, but since it is present in R* and the definition of R* it is that it is a union of all powers of R.

So, if a_i is related to a_j and R*, that means it is present in one of these powers of R say if ith power. Then as per this theorem statement, which we have proved in the last lecture there exist a path of length i from the node a_i to a_j. It might be possible that (a_i, a_j) is also present in say some other power of R. That is also possible say its present in the kth power. That means, as by the same statement there exist a path of length k from the node a_i to a_j.

So, that is why I am not focusing on the path length here. I am just stating here that if at all the element a_i is related to the element a_j, in the relation R*, then some path exists from the node a_i to the node a_j in your graph of the relation R. Now, we will be focusing on this connectivity relation where the relation R is defined over a finite set. That means R is defined over a finite set, consisting of n elements.

My claim here is that R* is nothing but the union of R, R^2 and up to R^n because you do not need to take the union of higher powers of R. Any higher powers of R will be subsumed in the union of the first n powers of R provided your relation R is defined over a finite set consisting of n elements. And this is because what can be the maximum path length between any two nodes?

Remember in the graph of your relation R, you have n nodes because now the relation is defined over a set consisting of n elements. So, what can be the maximum path length? The maximum path length can be n only because you have only n distinct nodes possible. Of course, you can keep on traversing along this path again and again that will be considered as a path of a higher length.

But, what do I mean by maximum path length? By maximum path length I mean here, maximum path length where path has distinct edges and why distinct edges? Because if I say for instance, consider R^{n+1}, that means I am interested to find out whether they are exist a path of length n + 1 between any 2 nodes in the graph G. Well, since I have only n distinct nodes possible, then the path of length n + 1 is possible only if a node is repeated in the path.

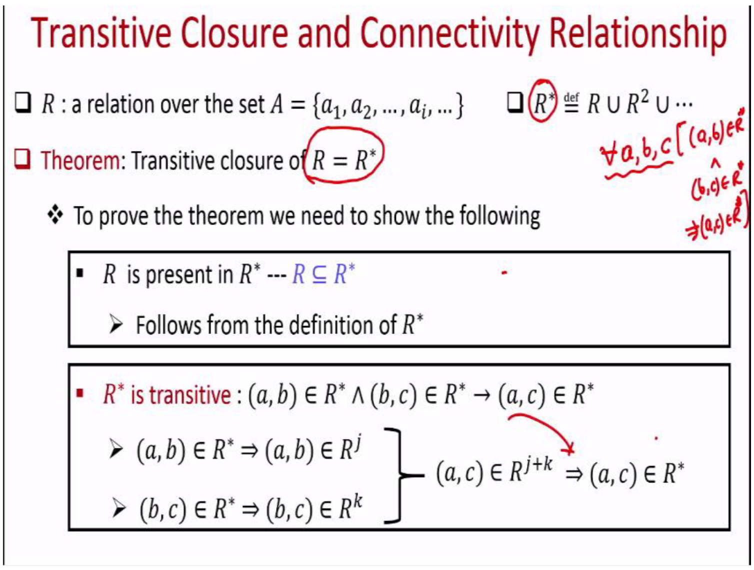
That means say for instance, I have a path of say, n is 3. Now say n equal to 4, So, I can say a path of length 5 exist between a_1 to a_2. Because I can go from a_1 to a_2 that is 1, and a_2 to a_3 that is 2, and a_3 to a_4 that is length 3, and then a_4 to a_1 that is 4. And then again from a_1 to a_2 that is length 5. But the same path can be considered as a path of length 1 because you have the node a_1 to a_2.

So, what I can say is that both (a_1, a_2) will be present in the relation R, and the same (a_1, a_2) will also be present in R^5, same (a_1, a_2) will also be present in R^9 and so on. So, when I will be taking, when I will be constructing R*, which will be union of R and R^4 then anything which is present in R^5 and R^9 and so on. Say for instance this element (a_1, a_2) which is present in R^5, R^9 would have been already subsumed in R itself. I do not need to separately consider those pairs by considering the fifth power of R and so on.

Because I know that, I cannot have a path of length more than 4 consisting of distinct edges. If at all there is a path of length more than 4, that means some edges and nodes are repeated. That means that path by excluding the repeated edges and nodes would have been already counted in some lower power of R when I would have constructed R*.

So, that is why when my relation R is defined over a finite set consisting of n elements. Then I have to construct the connectivity relationship I just need to focus on the first n powers of relation R. There would not be any extra ordered pairs beyond this n different powers of relation R which are present in R*.

(Refer Slide Time: 09:07)



So, we have not proved in detail but intuitively this is the statement. So, now what we are going to prove here is now we are going to see a relationship between the transitive closure and the connectivity relationship. So, remember in the last lecture we saw an example where we constructed the transitive closure of a relation. And there we had iteratively applied the process, the rule that if you have (a, b) and (b, c) in the expanded relation R or in the original relation R, then you add the element (a, c) and keep on doing this process till we do not need to add any extra elements of the form (a, c). So now what we are going to do is we are going to formalize that process by stating this beautiful result that the transitive closure of your relation R is nothing but the connectivity relation. Now to prove this theorem, we need to prove several things. The first thing that we have to prove is that relation R, the original relation R is present in your R*.

Because that is one of the requirements of transitive closures, that your original thing should be original relationship be present in the closure of that relation. But it is easy to see that the original relation R will be a subset of R*. Because R* is nothing but R union the higher powers of R. So, your original R will be definitely present in R*. So, all the ordered pairs which were there in R will be present in R*.

The second thing that we have to prove is that indeed this expanded relation R which is R* is satisfying your transitivity property. And for proving that this relation R* is going to satisfy transitivity property what we are going to do is we are going to show the following. You take any arbitrary (a, b) and (b, c) which are present in R*, then you have the guarantee that (a,c) is also present in R*.

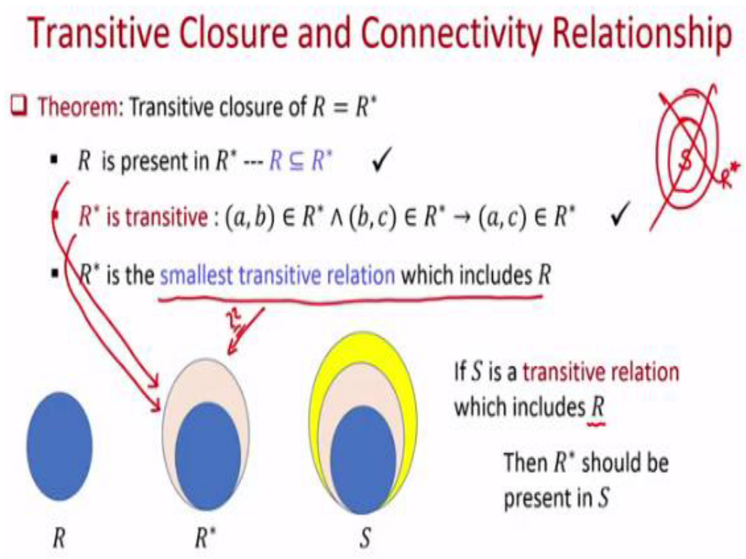
Why I am going to show it for arbitrary (a, b) and arbitrary (b, c), because the property of transitive properties, the transitivity relationship demand is that for all a, b, c. if (a, b) and (b, c) are there in your relation then you need (a, c) to be in your relation. And these need to be shown for all a, b, c. But we cannot take every possible a, b, c in R* and show this implication to be true. So, that is why we are showing going to show it for arbitrary (a, b) and arbitrary (b, c).

And then take the help of universal generalization and conclude that the statement is universally true for all the elements of the domain. So, assume to prove this implication we have to show that the left hand side of the premise of this implication is true and then we have to show that even the conclusion is true. That is how the definition of implication is given. So, assume that (a, b) is present in R* that means as per the definition of R*, (a, b) is present in some power of R, say the jth power because that is the definition of R*.

And in the same way imagine that (b, c) is present in R*, that means it is present in some power of R say the kth power, there may not be any relationship between j and k, j could be anything k would be anything. Now what I can say here is that tuple (a, c) will be present in the (j + k)th power of R. Because that is a definition of (j + k)th power, because (j + k)th power will be nothing but R^k composed with R^j.

That means you would have applied the relation R^j first, that means you will say that say (a, b) is there in R^j and on top of that you will apply the relation R^k. So, here b is acting as your intermediate element. So, (a, b) is present in R^j and (b, c) is present in R^k and we will conclude that (a, c) is present in the (j + k)th power of R. And (j + k)th power of R will be included in R* because that is the definition of R*. So, we proved this implication to be true for an arbitrary a, b, c that shows that the relation R* is transitive.

(Refer Slide Time: 14:01)



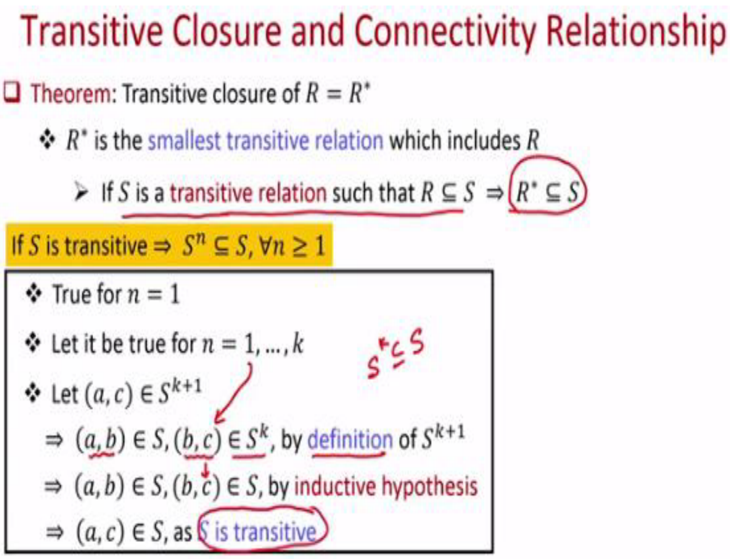
So, we have proved, we have shown that the two of the requirements of transitive closures are satisfied by your R* relation. Now we have to prove the important thing. We have to prove that

R* is the smallest possible expansion of your relation R which is transitive. And the way we are going to prove this is as follows. So, we have expanded our relation R to R*. We have shown that R is present in R* and we have also shown that R* is transitive.

Now we have to show this third property how we are going to do this. We will do this by showing that you take any transitive relation which includes R that means you take any expanded version of R which is transitive. In that expanded version of R namely S, R* is present. That will automatically show that R* is the smallest possible expansion. That means there is no smaller subset of R* which includes R as well as it is transitive.

That is what we are proving here pictorially. That means it is not the case that you have something of the following happening, that you have R* and you have say an expanded R say S which is transitive and which includes R, that is not going to happen, such that S is present in R* it is not going to happen. We are going to prove it other way around. We are going to show that you take any transitive relation, which is expansion of R, R* will be definitely present in that expansion S. And that automatically will show that R* is the smallest expansion that we have to do.

(Refer Slide Time: 15:55)



So, more formally we have to prove that you take any relation S, which is a transitive relation such that R is included in S. Then the expanded R namely the connectivity relation is included in this expanded relation S. So, to prove this property we are going to take help of a very small fact regarding the transitive relations and that fact is the following. My claim here is that if S is a transitive relation then you take any power of that transitive relation, it will be a subset of the original relation. And since this is a universally quantified statement for all n >= 1 we can quickly prove it by induction, the base case is obviously true because S is always a subset of S. Let the statement be true for n equal to 1 to k and now we are going to prove it for n equal to k + 1. So, imagine you have an arbitrary (a, c) element (a, c) present in the k + 1th power of S.

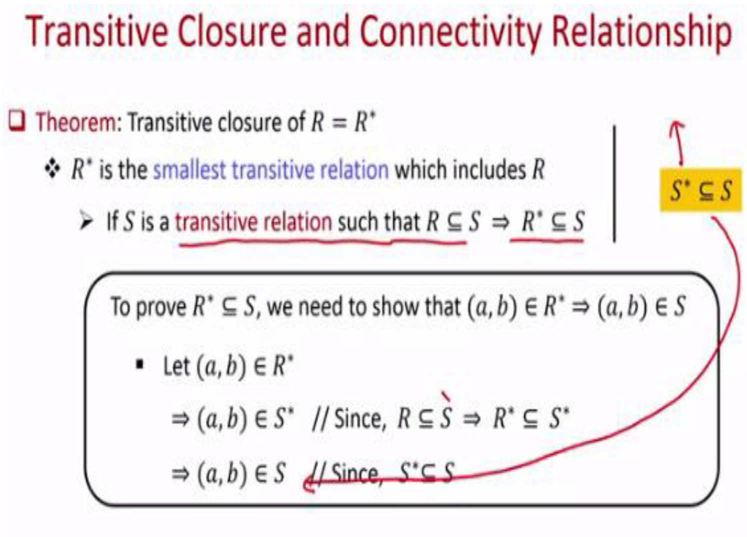
Then as per the definition of S^{k+1}, you have some intermediate element b such that a is related to b in the relation S and b is related to c in the relation S^k. Then I apply the inductive hypothesis here. Since (b, c) is present in S^k and statement is true for n equal to 1 to k. That means this (b, c) is present in S as well because S^k is a subset of S as per my inductive hypothesis.

Now, if I have (a, b) present in S; and (b, c) presence in S; I can say that (a, c) is also present in S. Because my base case or the hypothesis of the statement that I am proving is that S is transitive. So, this is a very straight forward fact regarding the transitive relations. If your relation is transitive you take any power of that relation, it will be always included in your original relation.

So, we have proved it for all n >= 1, now, I apply the definition of S*. What is S*? S* is going to be S^1 ∪ S2 ∪ S^n ∪ higher powers. So, each of this power is included in the relation S because that is what we have proved. We have to for all n >= 1, S^n is included in S. So, if you take the union of all the powers of S that will be included in S.

And that shows that the connectivity relationship satisfies the property that if your original relation is transitive then the corresponding connectivity relationship is included in the original relationship.

(Refer Slide Time: 19:06)

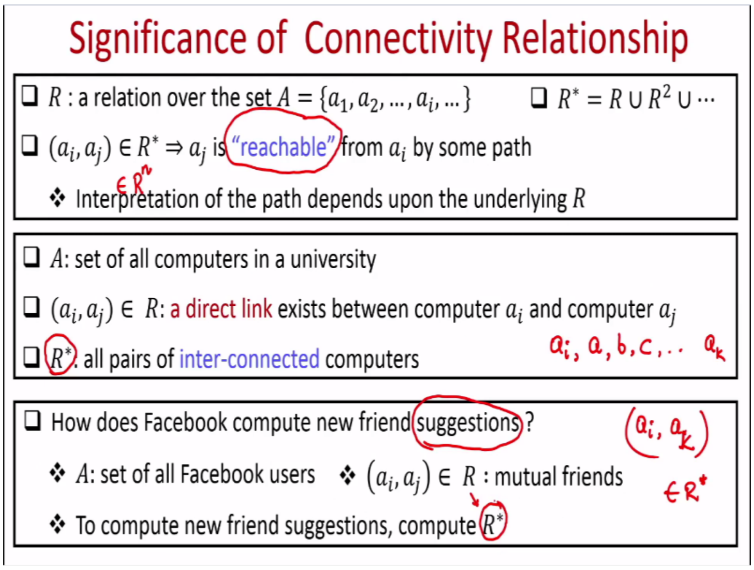


That is the side result which we are going to retain here. Now coming back to the statement that we want to prove here. We want to prove that if you take any transitive relation which is an expansion of R then the connectivity relation of R is going to be included in that expanded R. So, we have to prove this subset relationship property. So, the definition of subset is that if you have any (a, b) present in R*, I have to show the same (a, b) is present in S as well.

Provided the hypothesis of this implication is true. So, I take an arbitrary (a, b) and assume it is present in R*. Then since it is present in R*, it will be present in S* as well. This is because as per my hypothesis here, R is a subset of S. That means R^2 will be a subset of S^2 that also means R^3 will be a subset of S^3 and so on. You take any power of R that will be a subset of the corresponding power of S.

That automatically shows that R* is a subset of S*. So, if (a, b) is present in R*, it will be present in S* as well. But you are also given the hypothesis that S is a transitive relation. And if S is a transitive relation, then you take S*, it will be a subset of S itself. That means whatever is present in S* it is bound to be present in S as well. Hence we have shown that if (a, b) is present in R* it is present in S and we have proved the third requirement of the transitive closure as well.

(Refer Slide Time: 21:05)



So, this is a very important theorem which we have proved that which we have proved now. And the theorem is that the transitive closure of the relation is nothing but its connectivity relation. So, this connectivity relation has got significance here. So, let me show that significance. So, imagine you are given a relation R and R* is defined to be the union of different powers. So, if you interpret the connectivity relationship in terms of the directed graph representing your relation.

And what I can say is that a_i is related to a_j in this connectivity relationship, if a_j is reachable from a_i by some path. Because (a_i, a_j) present in R* means it is present in some power of R say the nth power, that means I have a path of length n from the node a_i to the node a_j. That is what we have proved and abstractly we can interpret it as if in the graph of your relation R the node a_j can be reached by some path from the node a_i.

Now the interpretation of this path depends upon what exactly is your underlying relation. So, for instance imagine A is the set of all computers in a university and I define a relation between two computers as follows. I will say that computer a_i is related to computer a_j, if there exists a direct link between computer a_i and computer a_j. That means a_i can directly send a message to the computer a_j by the channel or the cable through which the computer a_i is connected to computer a_j.

Then if I construct the connectivity relation R* for this given relation R then it is easy to see that R* will have all ordered pairs of the form (a_i, a_j) where a_i is connected to a_j may be directly or through intermediate computers. That means R* basically talks about all interconnected computers in your university. Now, let me give you another example of this connectivity relationship.

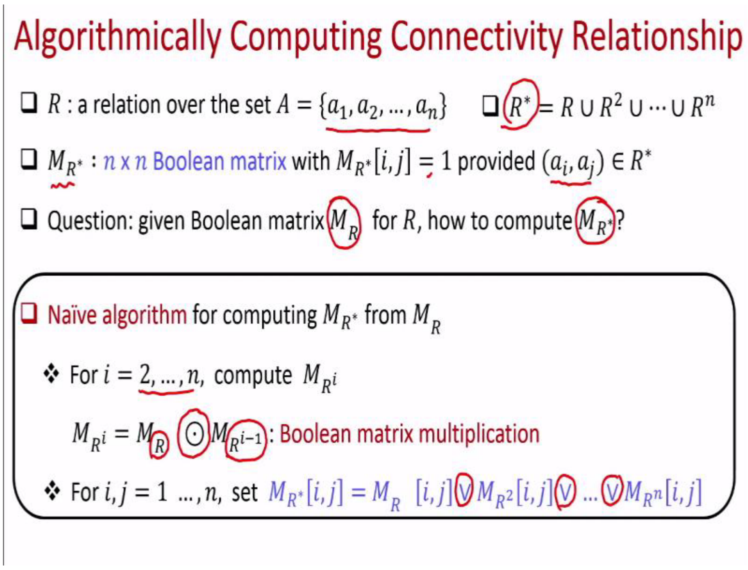
So, have you ever wondered that how Facebook computes new friends suggestions? So, imagine A is the set of all Facebook users and I define a relation between two Facebook users as follows. I call that relation as R. I say that user a_i is related to user a_j provided a_i and a_j are mutual friends over Facebook. Now, if you take the relation R* with respect to this R namely the connectivity relationship defined over the relation R over the Facebook users.

Then if you have a_i related to a_k in the relation R*. That means I can say there are series of a sequence of intermediate Facebook users such that a_i is related to the first user then the first user is related to the second user and so on and the second last user is related to a_k. That means in terms of a graph theoretic property, you can interpret that there is a path of some length from a_i to a_k.

That means even though a_i and a_k may not be mutual friends, what Facebook can think if that it might be the case that a_i has not searched for the Facebook user a_k and vice versa. So, let us send a suggestion to a_i that well, you know the user a_k or not and similarly to the user a_k. Because a_i is directly related to some a, the user a is related to user b, the user b related to user c and say there is by sequence of intermediate users you have a_i related to a_k.

And since friendship is by default considered to, be transitive. Of course, there might be exceptions as well, where a is friend to be b, and b is friend with c, but a need not be friend with c. That is why Facebook just give you a suggestion that hey looks like that we have found someone who is related to you by some intermediate users and to do that basically the Facebook algorithm has to construct here the connectivity relationship of this relation R.

(Refer Slide Time: 25:58)



So, this connectivity relationship has got a huge amount of significance. Now the question is how we algorithmically compute this connectivity relationship. So, we will be focusing on the case when the relation is over a finite set. And recall that in this case R* is nothing but the union of first n powers of R. So, what we are going to construct is, we are going to construct the Boolean matrix M_{R^*}.

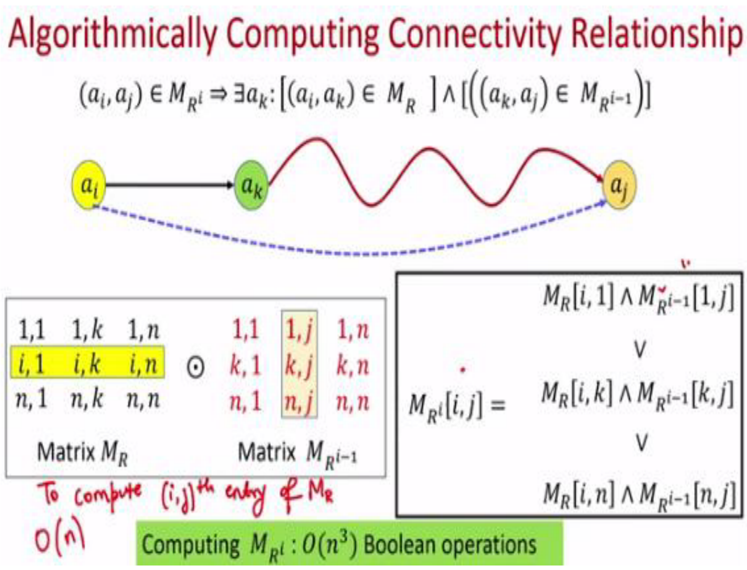
So, it will be an n x n Boolean matrix, representing your connectivity relation R* where the ith row and the jth column will be 1 provided the element a_i is related to a_j in the connectivity relationship R*. So, the question that we want to address here is that you are given the original relation R namely the Boolean matrix representing the relation R. How you compute the Boolean matrix representing the relation R* where the relation R is defined over a set consisting of n elements.

So, here is the naive algorithm for computing the matrix for connectivity relation. We compute the matrix for different powers of relation R. So, we are already given the relation for R. We are already given the matrix for the relation R. So, we do not need to compute that. So, that is why for i equal to 2 to n, we compute the matrix for the next powers of R. And this we can compute by performing a Boolean matrix multiplication operation, which I will discuss very soon.

And here what I am going to do is to compute the matrix for the ith power of the relation R. I am going to multiply the matrix for the original relation along with the matrix for the (i – 1)th power of the relation R provided. I have already computed it. And then what I am going to do is, I am going to take the Boolean disjunction of the individual n x n matrices for the different powers of R that I have computed.

And that will help me to get the Boolean matrix for this connectivity relation R*. So, now there are two operations here the Boolean matrix multiplication and the disjunction operation.

(Refer Slide Time: 28:29)



So, I am going to define them one by one. So, let us start with this Boolean matrix multiplication here. So, the goal here is the following. You are given the relation R or equivalently its matrix representation and say you have already computed the matrix representation for the (i – 1)th power of R. And now your goal is to compute matrix for the ith power of the relation R. So, recall as per the definition of the ith power of the relation R, a_i will be related to a_j, if you have a_i related to some a_k in the relation R, and a_k is related to a_j in the relation $R^{(i-1)}$, that is what is the definition here. So, that is what pictorially I have represented here. When that is the case, then you will say that a_i is related to a_j in the ith power of the relation R. So, basically this we have to check whether this structure is present in the graph of relation R or not.

If this structure is present then we can say that a_i is related to a_j. Now, how do we check this structure? So, we focus on the matrix for the relation R : M_R, and for the matrix of the, (i – 1)th power of R : M_R^{i-1}. And my claim is to check whether a_i is related to a_j or not. It is sufficient to take the Boolean dot product of the ith row of matrix M_R and the jth column of the matrix of (i – 1)th power of R : M_R^{i-1}. This is because if you take the Boolean dot product this (i, 1) will be multiplied here with the entry (1, j).

And multiplying here means conjunction; because remember the matrix M_R and the matrix M_R^{i-1}, both are Boolean Matrix. They just say whether something is related to something or not. So, I can say that if in my graph the element a_i is related to element a_1, and if element a_1 is related to element a_j. Then I can say that element a_1 is related to a_j. So, that is what is the essence of checking this first conjunction.

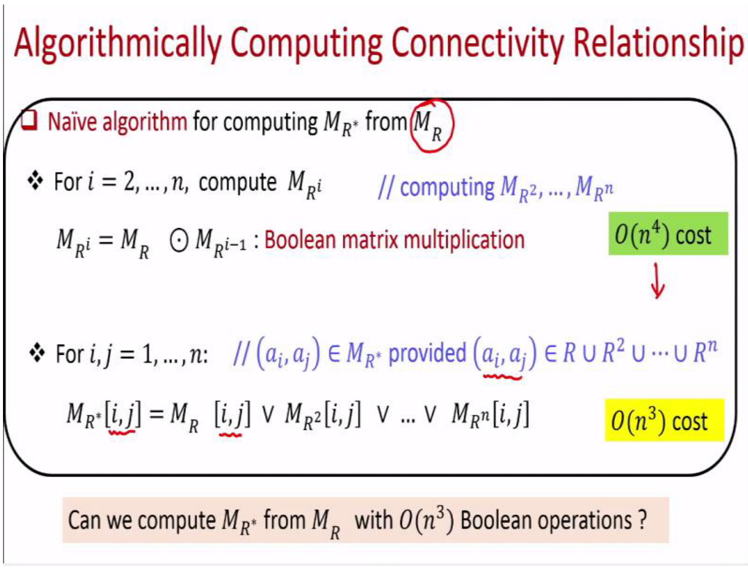
Checking whether (a_i, a_1) is related in the relation R or not. And whether (a_1, a_j) is present or related as per relation R^{i-1} or not. That is the case I do not care for the other expressions in this overall expression. I can simply say that the i, jth entry in R^i should become 1 or I should check in the similar way that a_i is related to say a_2. And a_2 is related to a_j or not. So, that will be the essence of second conjunction here.

In the same way I should check whether a_i is related to a_k in the relation R. And a_k is related to a_j in the relation R^{i-1} or not. And in the same way that will be the essence of this kth conjunction. And in the same way I should check whether a_i is related to a_n directly. And a_n is related to a_j or not in the relation R^{i-1}. If any of these n conjunctions hold, then I can say that definitely a_i is related to a_j in the relation R^i.

But if none of these n conjunctions are true then disjunctions of n 0s will be 0s and that is why the (i, j) th entry will remain 0. So, you can see here that if you are given the Boolean matrix of the relation R and if you have computed Boolean matrix for relation, R^{i-1} then to compute the, (i, j)th entry. So, let me write down. To compute (i, j)th entry of M_R, you need to perform O(n) Boolean operations.

Because you will be performing the dot product of two vectors of size n that will take you order of O(n) effort. And how many such (i, j) entries are there, that I need to compute in the matrix of R^i. Here are n^2 entries so, that is why it will be order of n^3 computation.

(Refer Slide Time: 34:16)



Now coming back to the Naive algorithm, the first step was computing the matrix for the different powers of R starting from the Boolean matrix of the relation R which is given as an input to you. So, that is why by applying the Boolean matrix multiplication operation that we have discussed just now we can compute the matrix for the individual powers of the relation R. And now what I have to check is I have to check whether a_i is related to a_j in any of these n powers of R or not.

And for that, I just need to check whether a_i is related to a_j in M_R^1 or whether a_i is related to a_j or not in the M_R^2 or whether a_i is related to a_j or not in the M_R^n or not. So, I have to just perform disjunction of n Boolean entries to find out the status of a_i, a_j in the matrix of R*. So, it turns out that the first operation here namely computing the matrix of different powers will cost me $O(n^4)$ effort, whereas computing the final matrix of R* will cost me $O(n^3)$ Boolean operations. So, now the question is can I reduce the overall cost here to $O(n^3)$.

So, we have this naive algorithm of computing the matrix for connectivity relationship, which cost me $O(n^4)$ efforts. My goal will be to do it with $O(n^3)$ Boolean operations, which we will do in the next lecture. So, just to summarize in this lecture, we introduce the connectivity relationship of our relation. And we saw that the transitive closure of any relation is its connectivity relation. And we discussed the naïve algorithm for constructing the connectivity relationship for a relation defined over a finite set. Thank you.