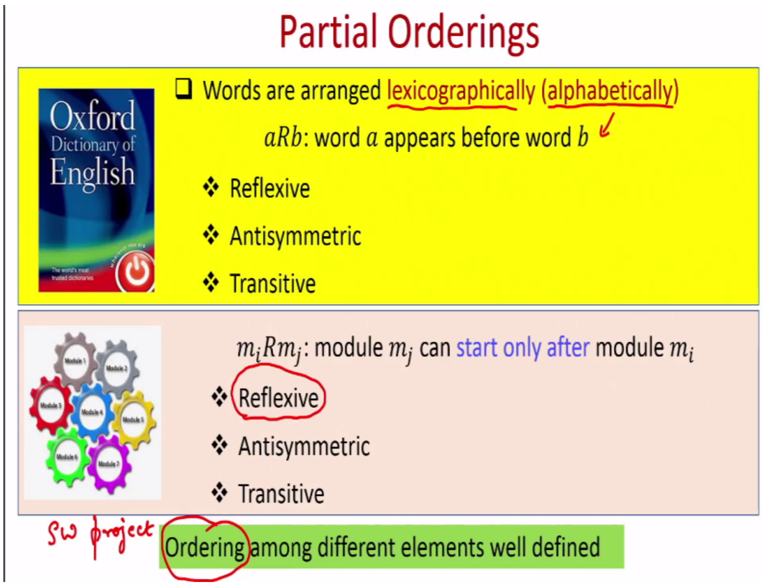
Discrete Mathematics
Prof. Ashish Choudhury
Department of Mathematics and Statistics
International Institute of Information Technology, Bangalore

Lecture -23
Partial Ordering

Hello everyone, welcome to this lecture on partial orderings. So, we will introduce the definition of partial ordering in this lecture. We will discuss Hasse diagram and we will compute with Topological sorting.

(Refer Slide Time: 00:35)



So, what is the partial ordering? So, if you consider a dictionary then the words in a dictionary are arranged alphabetically or we also say that the words are arranged lexicographically. And there is a very nice relationship which you can state that holds between the words or relationship holds with respect to the way in which the words are arranged in your dictionary. So, the relationship here is, I say that a word a in the dictionary is related to the word b in my dictionary provided the word a appears before the word b.

So, this alphabetical arrangement of the words can be considered as a relationship. And it turns out that this alphabetical arrangement of the words in the dictionary satisfies three properties. It satisfies the Reflexive property, it satisfies the Antisymmetric property and it satisfies the Transitive property.

Reflexive property because implicitly I can always say or I can always assume that a word always appear before itself. That is not true in the sense of the dictionary, but I can always have this implicit order. The alphabetical arrangement of the words satisfies antisymmetric properties because you cannot have two different words such that the word a appears before the word b and simultaneously the word b appears before the word a.

And this alphabetical arrangement of the words satisfies the transitive property because if you have the word b, appearing after the word a, and if you have the word c appearing after the word b, then you can say that the word c is appearing after the word a. So, that sense it is a transitive relation. It turns out that you can have several such relations which satisfy the property of being reflexive, antisymmetric and transitive.

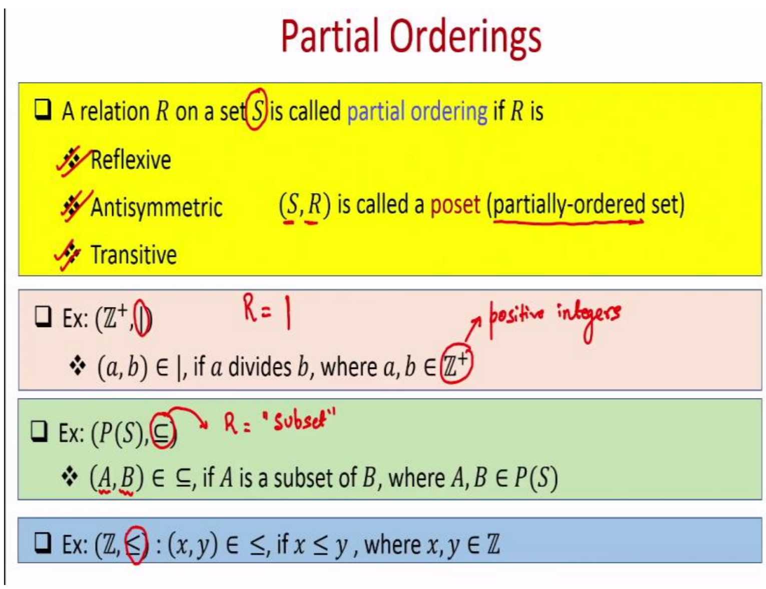
So, for instance imagine, I have a big software project. And typically in a big software project you identify various modules, various components which are independent of each other and each of them can be executed by separate procedure. So, now imagine that I have several such modules and I have defined a relationship or a dependency between the modules by a relation R and I say that module m_i is related to the module m_j if there is a dependency on the module j for the module i. So, I define a relationship R where module i is related to module j provided module j can start only after module i is over. That means until and unless you are done with the module i, you cannot start the module j. That is a dependency relationship. Now again I can say here that this dependency relationship, is reflexive, it is anti symmetric and it is transitive.

It is reflexive in the sense I can always simply assume that a module always depends on itself. It is an implicit dependency. This dependency relationship is anti symmetric because I cannot have two separate modules which are dependent on each other. If that is the case if that situation happens in your software project then it will lead to a state of a dead lock. So, for example, module 1 depends on module 2 and module 2 depends on module 1 you cannot start both of the any of them. So, that is why this relationship will be anti symmetric.

And is relationship is transitive. If module 2 depends on module 1 and if module 3 depends on module 2, implicitly it means that module 3 depends on module 1 as well. So, I have given you examples of two relations each of them satisfies the reflexive, anti symmetric and transitive properties and the essence of both this examples is the following.

You have a well defined ordering among different elements. If I take the first example my elements were the words of the dictionary and there is a way to ordering, the alphabetical order. If I take the second example, my elements of the set were the modules of the software project and there is a well defined ordering.

(Refer Slide Time: 05:22)



So, let us now generalize this theory. So, we now have we are now going to define a special type of relation which we call as a partial ordering. So, you are given a set S over which a relation R is defined and it will be called as a partial ordering, if the relation is reflexive, antisymmetric and transitive. In that case, the set S along with the relation R is called a poset. The full form of poset is partially ordered set.

Let me give you some more examples of partial ordering here. So, I consider the set of all positive integers, so this set $\mathbb{Z}^+$ is the set of positive integers. And I defined a relation divides which I am denoting by $|$. So, my relation R is the $divides$ ($|$) relationship.

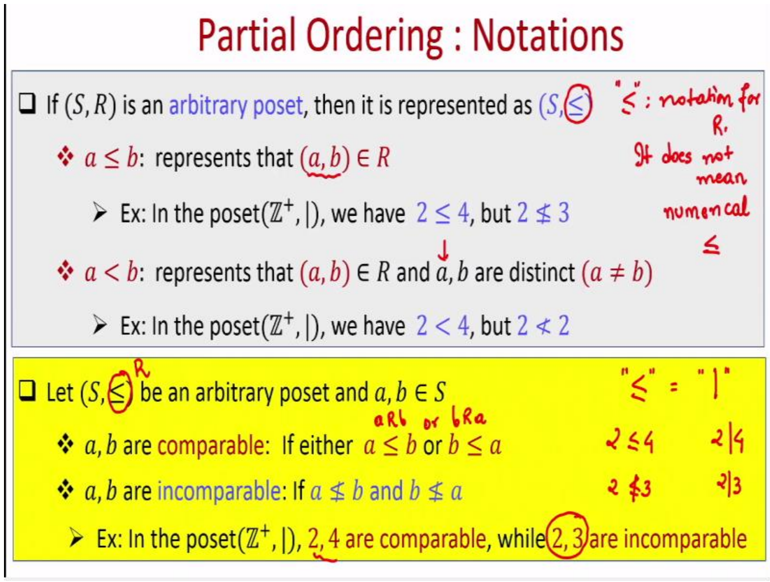
And I say that $(a, b) \in | \; if \; a|b$ and $a, b \in \mathbb{Z}^+$. Otherwise, a is not related to b. So, now it is easy to see that this relationship is reflexive because every positive integer divides itself. This relationship is antisymmetric, because you cannot have two different positive integers simultaneously dividing each other if a divides b as well as b divides a then that is possible only when both the integers are same.

And if $| \; if \; a|b$ and $b|c$, then you have $a|c$. So, it satisfies the transitivity property. So, this is an example of a partial ordering. Now, let me define another relation here my relation here is $\subseteq$. My R here is the subset relationship, which I am denoting by $\subseteq$. And my elements are the elements of the power set of a set. So, the relation is not defined over the set S. I stress that the relation is defined over the power set of S.

So, my elements are here subsets of S and I say that a subset A is related to the subset B, if$A \subseteq B$. That is my relation. Again this relation satisfies the reflexive property because $A \subseteq A$. It satisfies the antisymmetric property because you cannot have two different subsets A and B, $A \subseteq B$ and $B \subseteq A$, because that is the case that means $A = B$. And it satisfies the requirement of a transitive relation. If $A \subseteq B$ and $B \subseteq C$, then that means that $A \subseteq C$. So, this is an example of partial ordering.

Similarly, if I take the set of integers, $\mathbb{Z}$, and if my relationship is the less than equal to relationship where integer x is related to integer y provided $x \leq y$. Then again this satisfies the reflexive property, antisymmetric property and transitive property and hence this is an example of a partial ordering.

(Refer Slide Time: 09:00)



So, now if you are given an arbitrary poset instead of using the notation R for the relation, I use the abstract notation $\leq$. I Stress here that is $\leq$is just a notation. It is just a substitute for R, it is notation for R. It does not mean numerical less than equal to, that is very important. That means when I am writing $a \leq b$, that does not mean that a is numerically less than equal to b. That just mean, that the element a is related to element b as per my relation R.

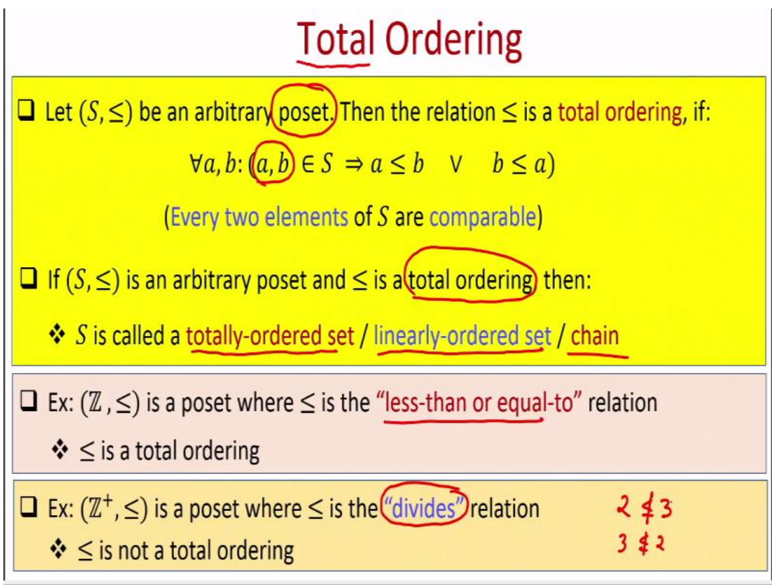
So, for example, if I take the partial ordering where my relation was the divide (|) relationship then $2 \leq 4$. Again, do not get confused by the numerical interpretation. Again numerically indeed 2 is less than equal to 4 but less than equal 2 here stands for the divide relationship namely $2 \mid 4$, but 2 is not less than equal to 3. Because, we are not numerically following the interpretation here, 2 does not divide 3. That is why 2 is not less than equal to 3.

Now we also use this abstract notation $<$ and again, this is not a numerical representation. It is just use to represent the fact that a is related to b but $a \neq b$. So, that is a case I use the notation a less than b. So, for instance, if I take the | relationship we have 2 less than 4, because indeed 2 divides 4, and 2 is not equal to 4, whereas we have 2 not less than 2, even though 2 is less than equal to 2, because 2 divides 2. But since 2 and 2 are same I cannot say that 2 is less than 2. So, that is the abstract notation that we are now going to follow for the rest of our discussion on partial ordering. Less than equal to is not numerical less than equal to, less than is not the numerical less than.

So, imagine you are given an arbitrary poset that so this less than equal to is an arbitrary relation R, which is a reflexive, antisymmetric and transitive. Now you take any two elements from the set S. They will be called comparable if $a \leq b$ or $b \leq a$, incomparable otherwise. So, again to demonstrate these two concepts let us consider the divide relationship, that means you are less than equal to relationship is the divides relationship.

Then you have $2 \leq 4$, because $2|4$. So, 2 and 4 are comparable. Comparable in the sense that they have there is definitely a relationship between 2 and 4. Either 2 is related to 4 or 4 is related to 2. But 2 is not related to 3, because 2 does not divide 3, that is why we will say that 2 and 3 are in comparable. So, in a partial ordering it is not necessary that you take any pair of elements and they are comparable. You may not have any relationship among them as per the relation R that you are considered.

(Refer Slide Time: 13:18)

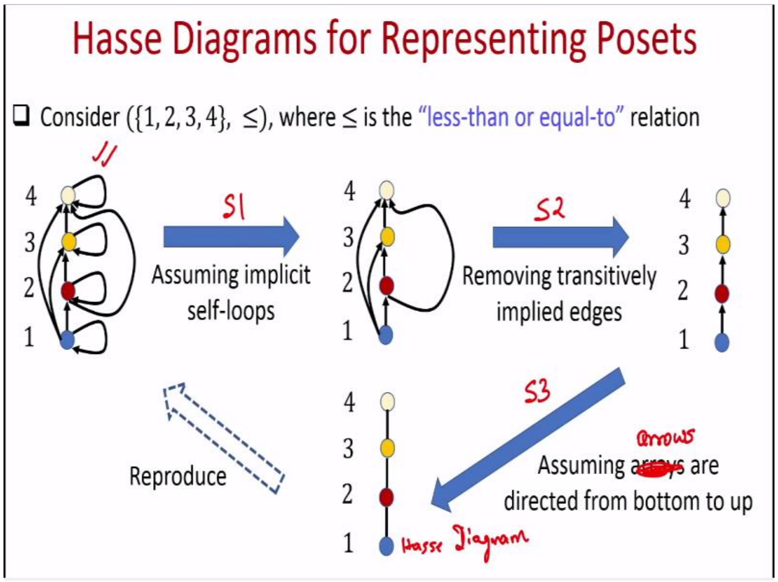


So, that brings us to the definition of what we call as a total ordering. And a total ordering is a special type of poset or partial ordering where you take any pair of elements from your set S, they will be comparable. That means either a will be related to b or b is related to a and that is why the name total ordering because you do not have any pair of incomparable elements. Whereas partial ordering the name partial denotes there, that you have ordering which is only partial. That means you may have a pair of incomparable elements and your relation R whereas a total ordering means you do not have any pair of incomparable elements. $\forall a, b \in S, a \leq b$ or $b \leq a$. So, that is why when your poset is a totally ordering, that means your relation is a total ordering then the set S is called as a totally ordered set. It is not called a partially ordered set.

In partial order set you might have the possibility of existence of incomparable elements. But in a totally ordered set you have relationship present between every pair of elements in the set. A total order set is also called as a linearly ordered set or a chain. Why it is called a chain or a linearly ordered set will be clear soon. So, let us see some examples of total ordering.

If I consider the less than equal to relationship namely, the numerical less than equal to relationship over the set of integers, then it is a total ordering. You take any pair of integers numerically either the first integer will be less than equal to the second integer or the second integer will be less than equal to the first integer. Whereas if you take the divides relationship, where a is related to b, provided a divides b then this is not a total ordering. Because 2 is not less than equal to 3 and 3 or also is not less than equal to 2. So, 2 and 3 are incomparable elements.

(Refer Slide Time: 15:38)



So, it turns out that we can represent partial ordering or posets by a very specific type of diagrams which are called as Hasse diagrams. And this is possible provided, this makes this is more interesting for posets where the relation is to find over a finite set. So, what exactly is this Hasse diagram? So, let me demonstrate this Hasse diagram with this less than equal to relationship which is the numerical less than equal to relationship defined over the set $S = \{1, 2, 3, 4\}$.

So, this will be the directed graph for your relationship less than equal to. Since 1 is related to 1, I have the self loop at the node 1, 2 is related to 2, so I have the self loop at the node 2. Similarly, I have the self loop at 3 because 3 is related to 3 and I have the self loop at 4, because 4 is related to 4. I have a directed edge from 1 to 2, because 1, 2 is present in the relationship. I have a directed edge from 1 to 3 because 1 is related to 3 and so on.

So, all the directed edge which are supposed to be present in the relation are there in this graph. Now what I can say here is that there is no point of explicitly writing down or stating the self loops. Because I can say that since my relationship is reflexive anyhow, I can always say that the self loops are implicitly present in my diagram. No need to unnecessarily represent them and make the diagram untidy.

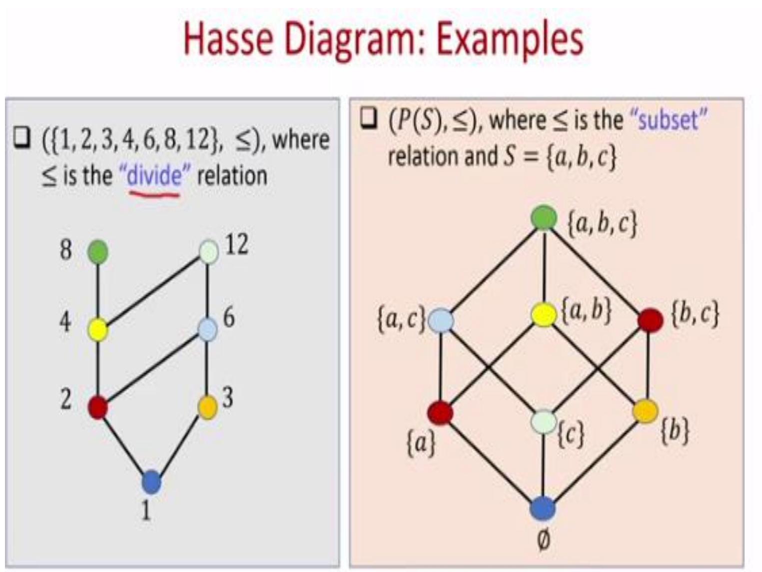
So, if I remove the self loops and assume that my, self loops are always implicitly present, then my diagram looks little bit better. Next what I can do is I can remove the transitively implied edges from this diagram and say that hey, since my relation is anyhow transitive, I can remove the edge present from the node 1 to 3. Because I can say that since 1 to 2 is present and 2 to 3 is present anyhow 1 to 3 will be present in my diagram. So, why to again explicitly represented in the diagram. So, I can remove all the transitively implied edges and my diagram simplifies further.

So, what I am doing is in each stage, I am trying to make my diagram more and more cleaner, tidy and try to remove all unnecessary information or redundant information, which I am not supposed to explicitly state in my graph of the relation of a partial ordering.

Now what I can say is that I can say that I make the assumption here that the arrays here, so sorry it is not arrays, this is arrows. So, I can make the assumption here that the arrows are always directed from bottom to up and that will take care of the direction of the edges as well and my graph becomes further simplified to this diagram. And now there is no more of information, which I can remove from this graph and say that it still represents my original relationship that means what I mean by that is if I take this graph which I obtained by step 1 and then followed by step 2 and followed by step 3 then if you give me just this graph I can reproduce the original graph. How can I reproduce the original graph?

As per my definition, I will say that arrows are always pointed upwards. Then as per my assumption the self-loops are always there. But and as per my assumption all the transitively implied edges are also there in my graph. That means if you give me the graph, this final graph which will be called as the Hasse Diagram here. If you give me the Hasse diagram here, I can reproduce the entire original graph for the Partial ordering that you were given here right. So, that is how you construct a Hasse diagram for partial order.

(Refer Slide Time: 20:13)

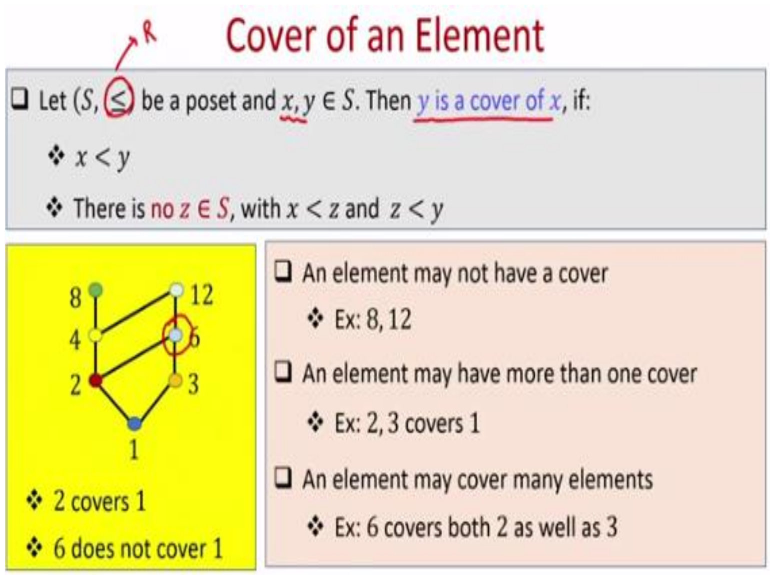


So, let us see another example here. So, you are given the divide (|) relationship. So, your ≤ is the divider relationship. So, again, we can start with our directed graph with the nodes 1, 2, 3, 4, 6, 8, 12, I can have all the self loops. I will have the transitively implied edges and so on and then if I remove all the self loops all the transitively implied edges, and if I remove the direction of the edges assuming that the arrows are always pointed from bottom to up, then this will be the Hasse diagram that I will obtain. This will be the minimum piece of information, which I need to retain in my graph to recover back the original diagram of ≤ or the | relationship over this set $S = \{1, 2, 3, 4, 6, 8, 12\}$.

Let us see another example where the less than equal to relationship is the subset relation. And relation is defined over the power set of $S = \{a, b, c\}$ not over the set {a, b, c}, remember. So, how many elements will be there in the power set of {a, b, c}? So, since set S has 3 elements the cardinality of its power set will be 2^3. There will be 8 subsets, $\{\phi, \{1\}, \{2\}, \{3\}, \{1,2\}, \{2,3\}, \{1,3\}, \{1,2,3\}\}$. So, these are the 8 subsets. Again, I have removed all the self loops. I have removed all the transitively implied edges and I have removed the direction of the edges.

So, for instance, I have not added the edge from the subset ϕ to the subset $\{a, c\}$, because that is transitively implied. Because ϕ is anyhow a subset of a which is represented by this undirected edge and undirected edge always have an implicit direction associated with it. And a is a subset of the subset $\{a, c\}$. Again, the direction is not explicitly mentioned here, but as per my assumption the directions are always upward. And as per my assumption the transitively edges are not explicitly stated in the graph. That means I have an implicit edge from ϕ to the subset $\{a, c\}$. Because indeed the subset ϕ is a subset of the subset $\{a, c\}$. But I do not need to explicitly add it in the graph. I can remove it. So, this is the minimum piece of information which I need to have in my graph to recover back the entire directed graph of the subset relationship over the power set of $\{a, b, c\}$. So, this will be the Hasse diagram of the subset relationship over the set $\{a, b, c\}$. So, why I am drawing all this Hasse diagram and all? Because that helps us to understand the next few concepts, which we are going to describe next.

(Refer Slide Time: 23:14)

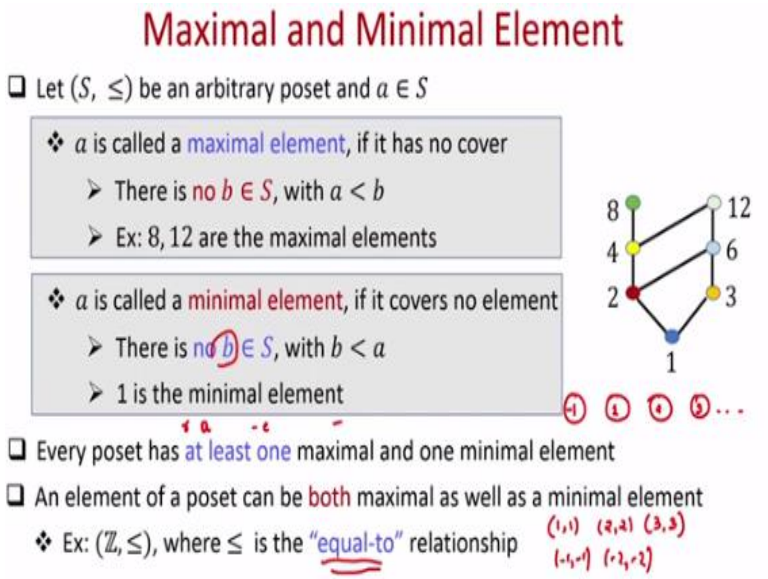


So, imagine you are given an arbitrary poset less than equal to relationship. This is not again a numerical less than equal to, this is an arbitrary relation, R, which is reflexive, anti symmetric and transitive. Then if I take a pair of elements x, y then the element y is called the cover of element x if the following two conditions hold. The element x should be related to the element y and of course $x \neq y$, that is why the less than symbol. And there should not exist any intermediate element $\exists z, x \leq z \; and \; z \leq y$.

So, pictorially, you can imagine that y is a cover of x if I view the Hasse Diagram then in the when I go from bottom to up y is immediately occurring or y is occurring on top of x layer wise and there is no intermediate element or no element z in the intermediate layer. So, for instance here in this Hasse diagram the element 2 covers the element 1 because in between 2 and 1 there is no intermediate element. You have the element 1 which is related to the element 2 and between 1 and 2 there is no intermediate elements. But the element 6 does not cover the element 1, even though the element 1 is less than 6, because element 1 is indeed the related to 6 as per this Hasse diagram. But in between 1 and 6 you have this element 3 such that 1 is related to 3 and 3 is related to 6. So, that is why 6 will not be considered as a cover of 1, but 3 can be considered as a cover of 1 because in between 3 and 1 there is no intermediate element.

So, it turns out that in a partially partial order set every element need not have a cover. So, for instance, if you take the Hasse diagram on your left-hand side the elements 8 the element 12, it does not have any common. There is no element on top of 8, there is no element on top of 12. Similarly, an element, we have more than one cover. So, as I said earlier both 2 and 3 covers 1. And an element may cover multiple elements. So, for instance here, in this Hasse diagram or in this poset 6 covers 2 as well as 3. So, these are the some of the properties of the cover of an element.

(Refer Slide Time: 26:01)



Let us next define what we call as the maximal and minimal element in a poset. So, if you are given an arbitrary poset and an element a from the set S. Then the element a is called as the maximal element if it is on the top most layer informally, or in a loose sense or if it has no cover. More formally, a is called maximal element, $\exists b \in S, b < a$ i.e., is no element b on top of a that means there is no element b such that a is related to b where a is different from b.

So, if I take this poset, 8 and 12 are both maximal elements. Because there is no element on top of a or no element b such that 8 is related to that b. There is no element b such that 12 is related to that element b or so. There is 4 will not be called a maximal element sorry and 6 cannot be called a maximal element. Why 4 cannot be called a maximal element. Because 4 is related to 8, there is something on top of 4 and so on.

Similarly, I can define what we call as a minimal element. So, an element is called as a minimal element if it occurs at the lower level of your Hasse diagram or in other words, it has no if it covers no element. Namely, a is called minimal element, $\exists b \in S, a < b$ i.e., there is no element b in your set S which occurs below a or such that b is related to a. So, for instance the element 1 here is the minimal element. Because there is no element b for the down 1 in your Hasse Diagram such that b is related to 1.

That tells you that why when we constructed the Hasse diagram, we assume that arrows are pointed from bottom to up. That helps us to understand these notions of maximal element and minimal element in an easy fashion. Now, it is easy to prove that if you have a poset over a non-empty set. I forgot to mention here over a non-empty set, then it has at least one maximal element and one minimal element.

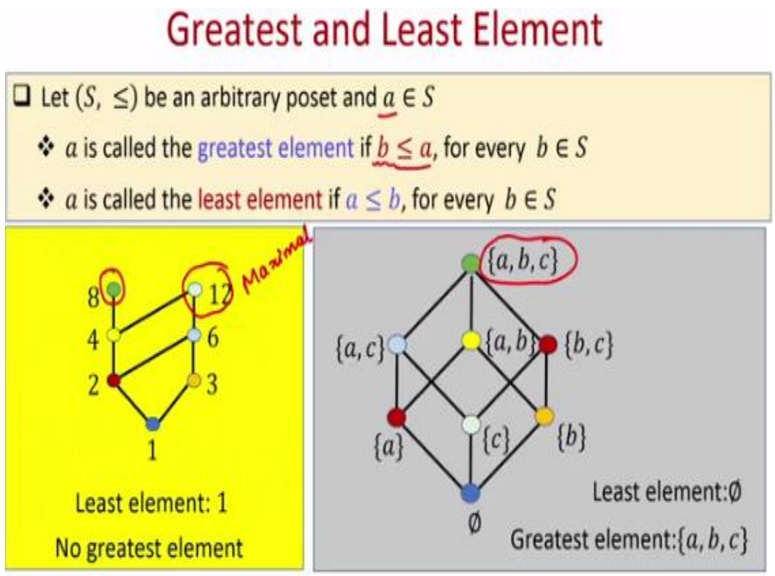
So, for instance if the poset is defined over a singleton element, then your Hasse diagram will be just a node itself say the element is a only. That means this is a valid Hasse Diagram representing the relation (a, a). And this relation is reflexive, anti symmetric and transitive and here the element a is both maximal element as well as minimal element.

Whereas if your set S has multiple elements and you will have a structure like Hasse Diagram and definitely there will be some element at the lowermost level and some element at the higher most level. So, those elements will be the maximal elements and a minimal. We can prove this thing formally but I am not going into that. Similarly, we can prove that an element of a poset can be both maximal as well as minimal. It is not necessary that the maximal element and minimal element should be different and there can be many maximal element many minimal element.

So, for instance if I consider the equal to relationship $(=)$ over $\mathbb{Z}$. Then I will have the elements of the form $(1,1), (2, 2), (3, 3)$ in my relation or the negative ordered pairs of the form $(-1, -1), (-2, -2)$ in my relation. What will be the Hasse diagram look like? The Hasse Diagram will just look like each integer within itself. So, the Hasse diagram will have no edges first of all. Why no edges? Because any element; is related to itself that is all. It is not a related to any other element.

So, the actual directed graph for this equal to relationship will have only self loops. It will have no other edges. And when we construct a Hasse diagram from that diagram, we will remove all the self loops. And as a result we will have a Hasse diagram where no edges will be present in my graph. So, in this graph all the elements are both maximal as well as minimal.

(Refer Slide Time: 30:21)



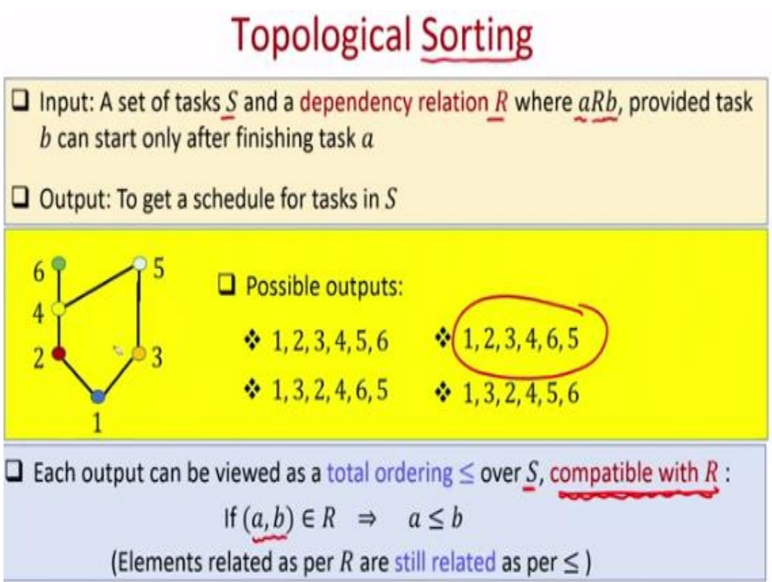
Now finally let us define what we call as the greatest element and the least element of a poset. So, if you are given a poset S and with an arbitrary relation less than equal to and if you have an element a then the element a of the set S is called as the greatest element if every element b is related to the element a as per the relation R or the relation less than equal to. In the same way the element a is called as the least element if it is related to every other element b as per your relationship less than equal to.

So, let me demonstrate these two concepts with this example. Here the element 1 will be the least element. Because you have 1 related to 2 you have 1 related to 3 you have 1 related to 4 even though the edge from 1 to 4 is not explicitly there, but as per the notion of transitive as per the definition of Hasse diagram all the transitively implied edges are there in my directed graph of the relation. Then similarly 1 is related to 8 and 1 is related to 6 and 1 is related to 12.

I cannot say that the least element is 2 because 2 is not related to 1 because the implicit direction of the edges are upwards. We do not have downward facing edges as in the Hasse diagram. So, the least element will be 1, but there is no greatest element. The elements 8 and 12 they are the maximal elements. But none of them is a greatest element, because there is no relationship between 8 and 12 and 12 and 8, they are incomparable elements here. So, I will have maximum elements, but that is not it is not necessary that I should have the greatest element present in my poset.

If I take this poset then here the least element is phi because phi is related to all other subsets as per the subset relationship. And the greatest element will be the subset $\{a, b, c\}$ because all other elements in this poset are related to this element $\{a, b, c\}$. So, if at all the greatest element exists in my poset, it will be unique but this is not necessary that a greatest element does exist in my poset. Similarly if at all these elements exist in my poset, it will be unique. But it is not necessary that every poset should have a least element.

(Refer Slide Time: 33:06)



Using all the concepts that we have discussed in, now we will now do a very interesting exercise here, which we call as topological sorting. So, in this topological sorting, you are given a set of tasks which is denoted by S and you are also given a dependency relationship R defined over the task in the set S and task a is related to task b provided b can start only after finishing the task a.

And what we want here is we want to get a schedule according to which we should finish the tasks in given in the set S. That means we have to decide which task to finish first and then which start to finish next and so on provided I have the dependency among the tasks given in the form of this relationship R. So, it is easy to see that the dependency relationship here is a partial ordering which can be described by a Hasse diagram. So, I am taking here a collection of task 1, 2, 3, 4, 5, 6 and a dependency relationship is given like this and what I want here is a schedule for scheduling the various task in the set S.

So, there can be multiples schedules possible. So, I have listed down four of them. I can finish the task one first and then I can finish the task 2 and then the task 3 then the task 4 then the task 5 then the task 6 that is one way of satisfying the requirement. Because once I am done with task 1 the dependency is over one and now I can freely choose either to do task, 2 or task 3. So, if I decide to task if I finish to, decide task 2, then next I can decide to either finish task 3 or I can decide to finish task 4. So, depending upon in what sequence I follow I choose the next task to complete that will give me for different possible schedules. So, it is not the case that there is only one possible schedule here. There can be multiple possible shapes here.

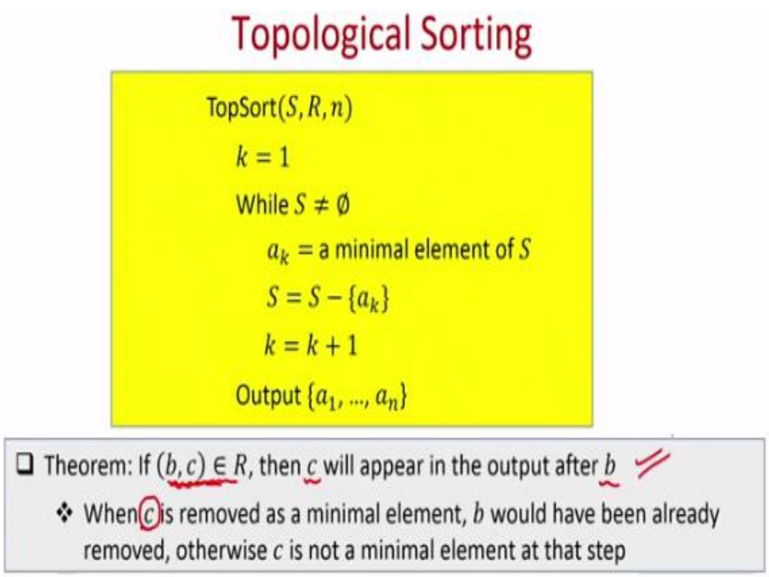
Now each of the possible outputs that I have stated over can be viewed as a total ordering over the set S compatible with my relation R. And why total ordering? You see in my original dependency relation, there are incomparable elements. So, for example, neither 2 is less than 3 nor 3 less than equal to 2 because they are not dependent at each other. Both of them depend on 1. As soon as I finish one I can freely decide or I can freely choose either the task 2 or the task 3. There is no dependency between the tasks 2 and 3 that is why it was only a partial ordering.

But if I say that my final schedule is this then in this final schedule is this then in this final scheduling I am saying explicitly that 2 is related to 3. That means I should I have finished 2 and then I have finished 3. So, in some sense this output sequence, which I have obtained here one of the possible output sequence of that I have obtained here can be considered as a possible total ordering on the task 1, 2, 3, 4, 5, 6 compatible with the relation R.

What do I mean by compatible with the relation R? By compatibility, I mean that if at all there was any dependency between a and b, that means if a was dependent on, if the task a was related to task b as per the dependency relationship. Then in the final sequence which I have obtained in the final total ordering which I have obtained it still the case that a is related to b. That means if in my Hasse diagram, if I was constrained to start task number 2 only after finishing task number 1, then in the resultant output sequence that constraint should be satisfied, it should not be validated. It is what I mean by compatible with my original relation R.

And you can see that each of the sequences which I have obtained here. In each of the sequences, I am satisfying the constraints which were given with respect to the original relation. None of the dependency which was maintained which was mentioned in my original relation R is violated in any of the output sequences which I have listed down, on any of the schedule which I have listed down.

(Refer Slide Time: 38:06)



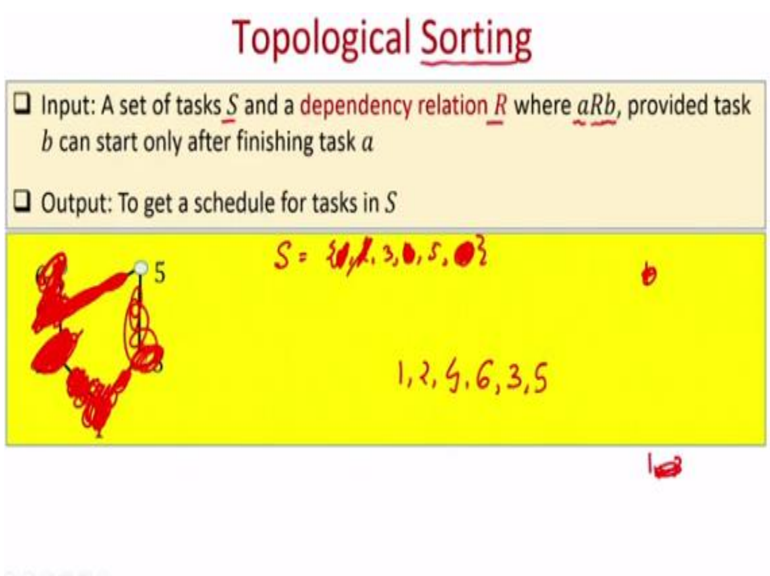
So, the general goal of the topological sorting is the following. You will be given a relation over a set S that relation may or may not be a total ordering. There may be incomparable elements present as per the relation R. What you have to output you have to output now a total ordering over the set S and the total ordering should respect the original relation R. It should be compatible with the original relation R.

That means whichever pair of elements which were related as per the original relation R, they should be still related as per your new ordering. It should not happen that a was related to b in the old ordering but in the new ordering a is not at all related to b. That should not happen. For the incomparable elements you are free to do whatever you want. But the elements which were comparable as per the original relation R, you have to maintain those that comparability property in the new ordering as well. That is the goal of Topological Sorting.

So, how do we do this? How do we output one such total ordering? So, the algorithm is as follows. We start with k equal to 1 and I will iteratively do the following till my set $S \neq \phi$. As soon $S = \phi$, I will stop my algorithm. So, what I will do is I will start with the minimal element that is there in my set S. And I will list it down; that means I have taken care of that element a_k in my total ordering and I remove that element a_k from the set S.

So, using my set S keeps on getting updated and that is why I start with my original set S here and every time in each iteration, I will be removing the current minimal element of the current set S and I will update the set S. And I will increment k to the next value of k and I do this till $S = \phi$. And once my set is becomes empty, I will list down the elements in the order in which I have removed them in this value. So, let me demonstrate this algorithm with this example here.

(Refer Slide Time: 40:27)



So, you have $S = \{1, 2, 3, 4, 5, 6\}$. Your $k = 1$. I start with the original S and find out the current minimal element. And in this case, I have only one minimal element namely the element 1. So, I will write down the element 1 to be the first task which should be taken care in my schedule and then I am removing the element 1 from my set S. That means the task 1 is taken care.

So, you can imagine that since task 1 is taken care there is no dependency of other tasks on the task number 1 and hence these two edges vanish from my Hasse diagram. Now, I have to find out the minimal element of the updated S. And I have two possibilities here. Both 2 as well as 3 are the minimal elements for the updated S and it is up to me. I can either choose 2 in my schedule to be the next task or I can choose 3 to be the next task in my schedule.

It is up to be the algorithm does not say that you have to if you have multiple minimal elements which one to choose. So, suppose I decide to take care of task 2. So, $k = k + 1$. So, I am not writing down the values of k here. Since I have taken care of the task 2, that means this task has vanished now. And now I have to choose the next minimal element and my minimal element are 4 as well as 3.

So, in my sequence, I can either put 4 or I can put or I can put 3, it is up to me. So, suppose I choose 4. So, sorry, so I am following the order 1, 2, 4 because I am taking care of 4 here. So, if I take care of 4, I am left with this set S. And now what is the minimal element? I can choose task 6 as well as task 3, because since 4 is also taken care, this edge also vanish. So, my minimal elements are element 6 and element 3. 5 is not minimal because 3 is related to 5. So, 5 is not minimal here.

So, it is up to me whether I put task number 6 or I can put task number 3. So, if I put task number 6 here, then 6 is taken care then I am left with only 2 task here and my minimal element is now only 3. So, I have to take care of the task number 3 and then finally I have to take care of the task number 5. So, that is essence of this algorithm.

In every step you are finding the minimal element, which is there in your updated set S put it in the sequence and remove it from your Hasse diagram. A very simple algorithm. So now we have to prove that the resultant output which we obtain from this topological sorting will be compatible with the original relation R. That means if at all the element b was related to element c or the task b was related to task c in the original relation, then even in the new sequence or the ordering that you have output the element c or the task c will appear after the task b. And that is very simple to prove.

The proof follows from the fact that when you were removing the task c from the Hasse diagram and putting it in the sequence, at that time it was the minimal element. Because in each step you only decide or you only choose to remove the minimal element from the updated set or updated Hasse diagram.

So, when it was the turn to remove the element c, at that point of time the element c was the minimal element in the Hasse diagram. That means at that point the element b would have been already removed from your Hasse diagram. If element b is still present in the Hasse Diagram, you have not removed it yet then you would have removed element b instead of element c because as per your original relationship $b \leq c$. So, no where you would have retained b and removed c, because in your original Hasse Diagram b was occurring on a lower level than c. And that is a very simple fact based on which we can state or throw this term.

So, that brings me to the end of this lecture just to summarize in this lecture we introduce the notion of partial ordering. A partial ordering is a relation, which is reflexive, anti symmetric and transitive. We introduce the notion of total ordering, Hasse diagram and we also saw the algorithm for topological sorting. Thank you!