INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

NPTEL

NPTEL ONLINE CERTIFICATION COURSE

An Initiative of MHRD

VLSI Design, Verification & Test

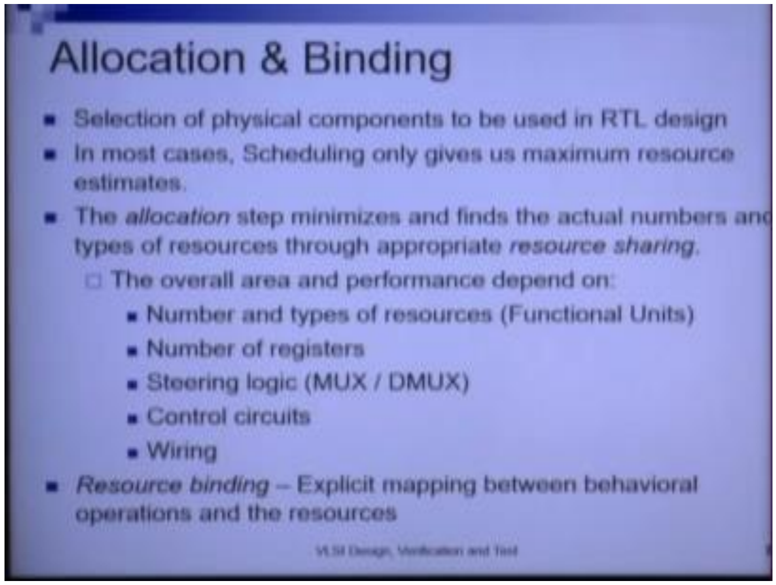
Dr. Arnab Sarkar

Department of CSE

IIT Guwahati

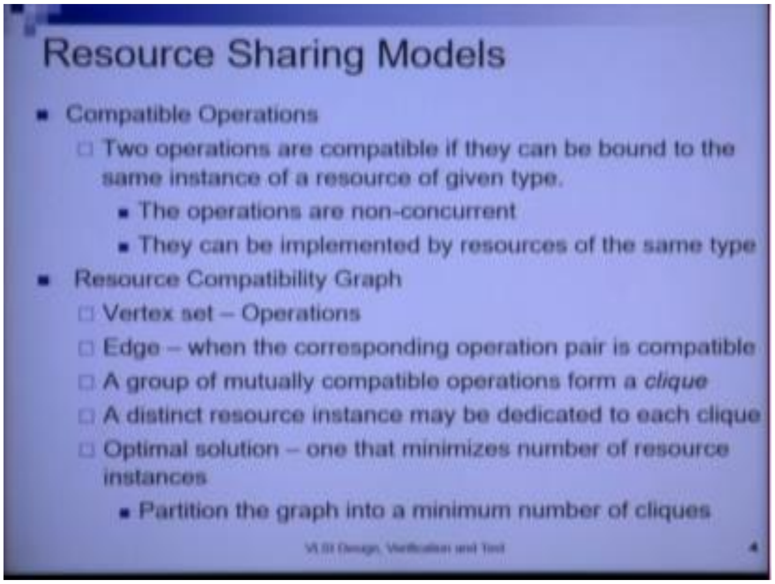
Welcome to module 2 of lecture six in the last module we had started with allocation and binding an important post scheduling step in high level synthesis in this lecture we will proceed with allocation and binding further.

(Refer Slide Time: 00:42)



Before taking a look at allocation and binding algorithms we will first take a look at different resource sharing models.

(Refer Slide Time: 00:51)



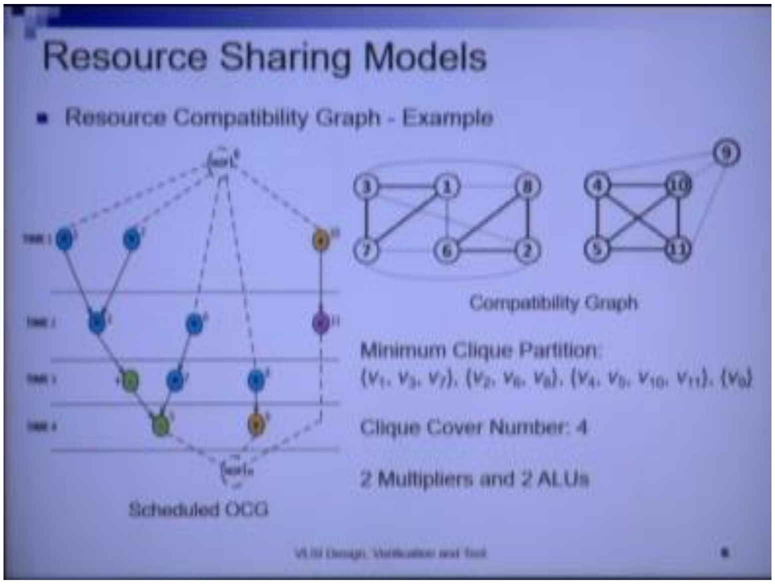
We look at a few definitions first we will understand what are compatible operations to operations are compatible if they can be bound to the same instance of a resource of given type that means when does two operations become compatible when the same functional unit can be used to execute both these operations when can the same functional unit instance be used when the two operations are non-concurrent if they are being execute if they are executing at two different time steps only then can I use the same functional unit to execute both these operations on the same functional unit.

And obviously they have to be implementable by resources of the same time now with the definition of compatibility we will take a look at two types of resource sharing graphs so first is the resource compatibility graph the other will be resource conflict of these two resource sharing graphs we will understand so what is a resource compatibility graph in the resource compatibility graph the vertex set of operations there is an edge between two operations if these two operations are compatible which means that there will be an edge between these two operations if both these operations are not scheduled at the same time step and both these operations can be executed by an operator of the same time.

A group of mutually compatible operations then form a click here what is it click a click is a max complete sub graph alright so a me two a set of mutually compatible operations form a click or a complete sub graph why because all these operations will be connected by edges between each other and white art that so because all these operations are mutually compatible either of these operations are scheduled at the same time step and the same functional unit type can be used to schedule all the operations in this clique a distinct resource instance can be dedicated to each click an optimal solution to the resource to the resource sharing problem is what.

One that maximizes resource sharing so the optimal solution you will be one that minimizes the number of resource instances so how do we minimize the number of resource instances if we can minimize the total number of clicks for each click I must dedicate a single of resource instance and if we can minimize the total number of clicks I will be able to minimize the total number of resource instances required. So partition the graph into a minimum number of clicks and obtaining this minimum number of clicks is to obtain the clique covers number, so the clique cover number of a graph is this minimum number of clicks into which the graph can be partitioned into.

(Refer Slide Time: 04:07)

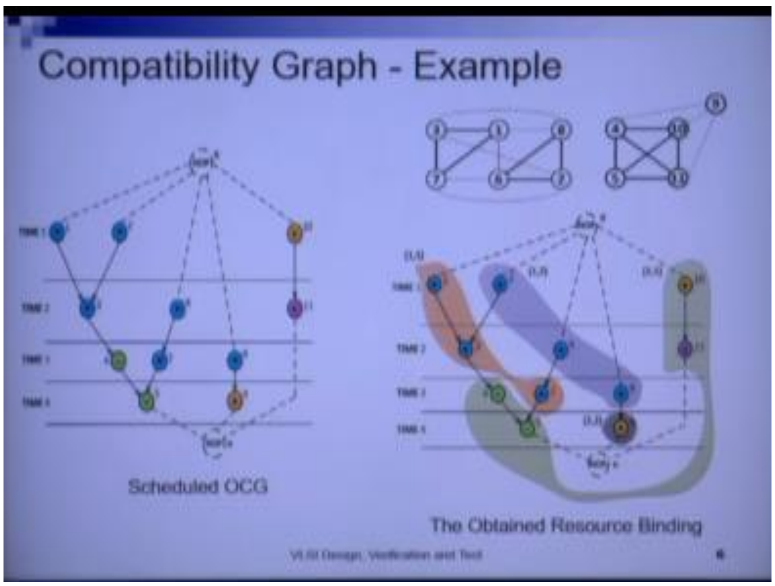


Now in this is a scheduled operation constraints graph on the left and on the right we have the corresponding compatibility graph in this graph we see that operation one and operation three share an edge why because operation one and operation three can be implemented by the same resource type multiplied and they are scheduled in different time steps one and two in this case similarly one and seven are also compatible why because one and seven are two operations that and we implemented by the same type of function unit Multiplan again and they have been scheduled in different time steps here time step 1 and time step 3.

Likewise all operations which share an edge are compatible one important thing to note here is that the compatibility graph will have as many disconnected components as there are resource types so here we have two resource types multiplier and ALU so for ALU I have a single connected component and another connected component for the x rated type now what are the minimum number of clicks what is the minimum click partition in this graph we have here v1 v3 v7 v1 v3 v7.

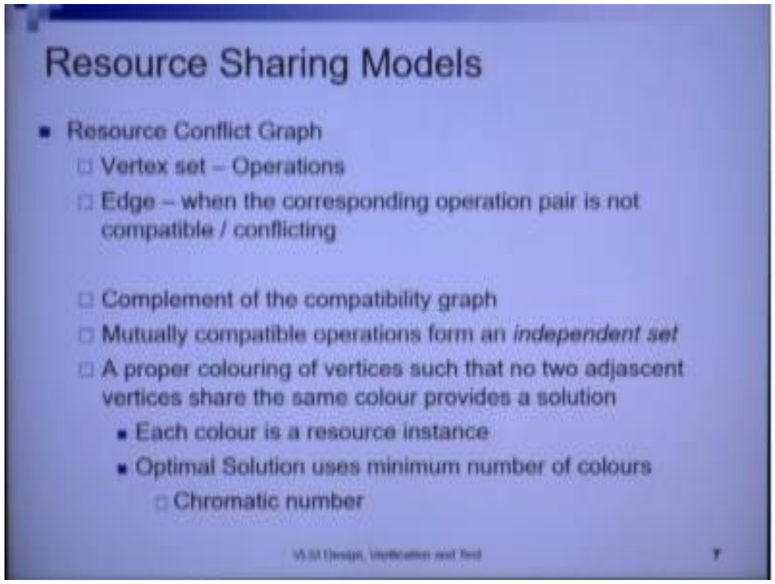
So the one with emboldened edges so these complete sub graphs with emboldened edges form the click so v1 v3 and v seven former click v6 v8 and V to form another clickv4 v5 v 11 and v10 form another click and v9 forms another separate click so these are the mean this is the minimum click cover that we have so the minimum click cover has four clicks and in indeed we require two multipliers and 2 ALU for the scheduled operation constraint graph.

(Refer Slide Time: 06:13)



Now as an example of the actual resource binding that we can obtain here is this so we have a 1multiplied say let us say multiplier 1multiplied instance one is used to schedule operation 137 which is in the same click again another multiplier say multiplied to is used to execute operations to 6 and 8 in time steps 1 2& 3 similarly a lu1 is used to execute the operations 10 11 4 and 5 in four different time steps and operation v9 is given another dedicated Lu for its execution. So this is an example allocation and this is the best possible allegation as it turns out.

(Refer Slide Time: 07:17)



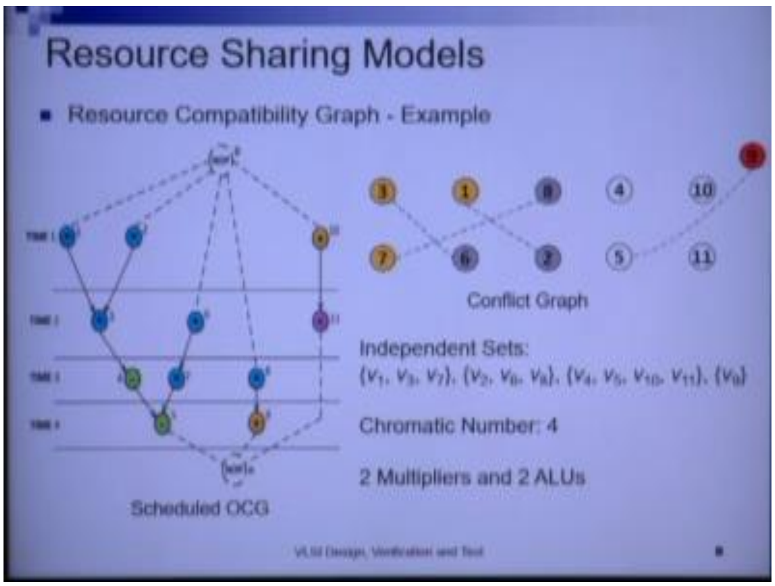
The second resource sharing model is the resource conflict graph resource conflict graph has the same vertex set as the compatibility graph so it's this its vertex set are also the operations of the operation constraints draft what are the edges the there is an edge between two operation between two operations if they are not compatible that means in the compatibility graph two operations shared an edge if they were compatible in the conflict graph to operation share an edge if they are not compatible.

Hence the conflict rock is a complement of the compatibility graph so when will too when will two operations be conflicting when these two operations are scheduled either in the same times step or these two operations are unrelated in meaning that they cannot be executed using the same functional unit type however it is true that if you have a set of compatible operations they will form an independent set what is an independent set the vertices in an handset will be those which do not share an edge between them.

So mutually compatible operations will form an independent set a proper coloring of vertices such that no two adjacent vertices share the same color provides a solution so what do the vertex coloring do vertex coloring tries to find the minimum number of colors that are required to color the vertices of a graph such that you cannot assign the same color to two vertices that share an edge these colors actually they denote resource instances each color is a resource instance.

And optimal solution uses the minimum number of colors such minimum number of colors is called the chromatic number of the graph example of a conflict graph.

(Refer Slide Time: 09:26)

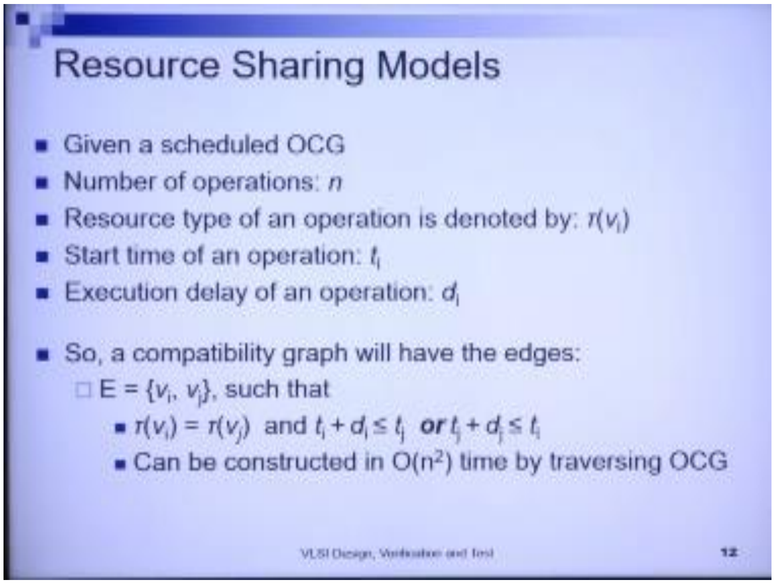


So in this we see that3 and 6 share an edge 3 and 6 share an edge why because 3 and 6 our operations of the same type and being scheduled in the same time step because they are scheduled in the same time step they are conflicting and cannot be executed by the same resource instance and therefore they share an edge obviously 3 and 4also share an implicit edge which is not shown because 3 and 4 our operations unrelated operations which must be executed by operators of different type there is an implicit edge between them which we have not shown here.

Now what are the meat meaning independent sets in this car conflict graph 37 and one is an independent set they do not share an edge between them to six and eight to six and eight is another independent set they do not share an edge between them for 10 5 and 11 is another independent set and v9 forms another independent set it is important note here that the that the clicks mean the compatibility graph are now independent sets in the conflict graph the chromatic number here is same as the vertex cover number in the compatibility graph is four and therefore we are going to multiply and to ALU allocation and binding will again be same.

Now one of the important outcomes of this analysis steed study of these models is that given general compatibility and conflict graphs both the minimum click partitioning problem and the minimum graph coloring problem are np-complete and hence exhaustive or an enumerative type so solutions only do exist for the general versions of the compatibility and conflict crafts now with this understanding of resource models we will first we will now see how from the scheduled operation constraint graph can be obtained a compatibility graph.

(Refer Slide Time: 11:56)



Similarly we using a similar procedure we can easily obtain the conflict graph as well because as we have seen that the conflict graph is just a complement of the compatibility graph now given a shade revolt OCG let n be the number of operations resource type of an operation is denoted by τ V_I it his comes from the scheduling problem so τ Vi tells me the type of resource that I require to schedule operation τ V_I the start time of the operation is τ V_I because this is a scheduled operation constraint graph I know the distinct start time for each operation I.

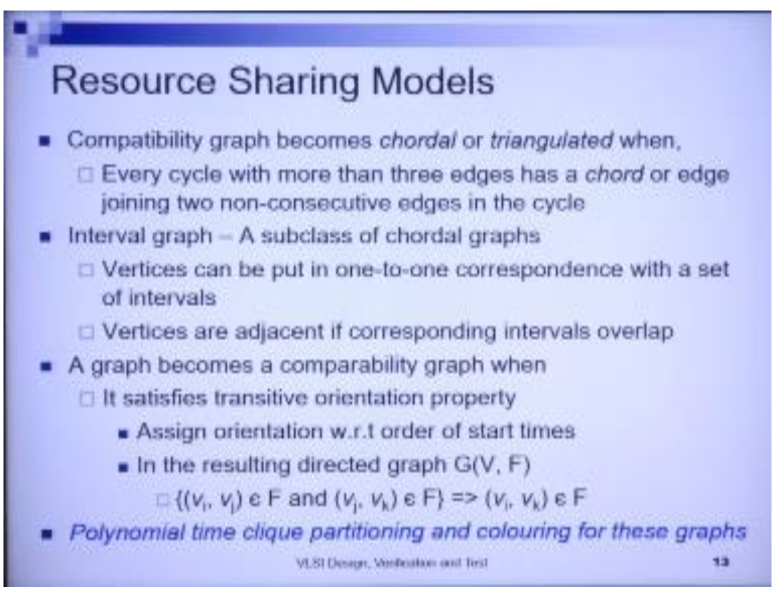
he execution delay of operation VI is also known this is d_i so the compatibility graph will have edges e = V_I V_J such that t V_I = t V_J the first here first term tells me that both TV I and τ V_I and τ V_J must be same that means they must V_I and V_J must be executable by the same type of resource the second constraint tells me that if T_J is scheduled later than T_I then T_J must be scheduled at least d_i time steps later than T_I starts so T_J start time must be at least di time steps after T_I why this is.

So because di is the delay of operation I operation V_j follows V_I and hence otherwise there is an overlap in the execution intervals of these two operations and therefore the same operation instance cannot be used to execute both these operations V_I and VJ similarly if if V_I starts after v-j then V_I must start at least D_J time steps after T_J after v-j starts so TJ plus DJ must be less than or equal to TI only then can I ensure that both operations VI + VJ will not overlap in that execution intervals and hence both these operations V_I and V_J can be implemented by the same type of resource.

Now we understand that how can we obtain these edges I need to traverse the operation constraints graph and then I need to consider each operation in the scheduled operation constraint graph then I find out which operate for a given operation in the operation constraints graph which are the operations that are adjacent to in the operation constraints graph for all those operations if this constraint is true I can put an edge between these two operations in the compatibility graph.

Now through such a traversal of the Opera operation constraint graph we understand that all edges of the compatibility graph can be obtained in Big O of n square time where n is the number of operations in the operation constraints graph.

(Refer Slide Time: 15:33)

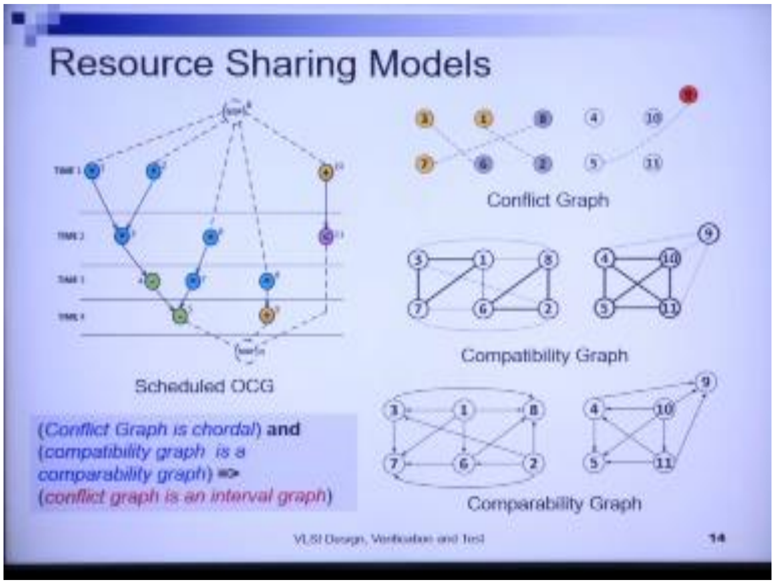


Now as we said for general compatibility and operation concern for general compatibility and conflict graphs both the minimum click partitioning problem and the graph coloring problem is np-complete and hence must be solved by enumerative methodologies however there are special classes of graphs for which for which both the minimum click partition in problem and the graph coloring problem can be solved optimally in polynomial time.

So if my compatibility graph or my conflict graph falls into one of these categories of these special types of graphs then my graph coloring problem or my clique partitioning problem can be solved optimally in polynomial time hence we need to understand these special types of graphs first to understand whether my graph coloring problem and the clique partitioning problem can be solved in polynomial time or not.

So first we understand what are coral or triangulated graphs if we say it in terms of a compatibility graph a compatibility graph becomes a choral or triangulated graph when every cycle with more than three edges in the graph has a cord or edge joining two non-consecutive edges in the cycle so I am given a compatibility graph and Ito find out is this compatibility grab a choral graph how do I find out if I see that each cycle of more than three edges in my compatibility graph has a cord joining non-consecutive edges in the cycle then I can say that my compatibility graph is a chordal graph.

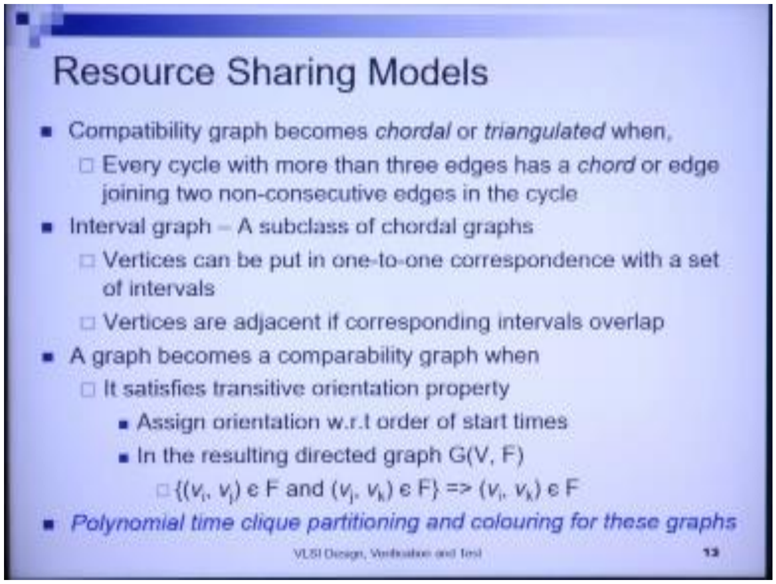
(Refer Slide Time: 17:43)



For example let us take this compatibility graph that I have here in this compatibility graph let us take one cycle 1376 in this cycle we say that there is a cord 17 here this cord this is a cycle of length more than 2 this is a cycle of length 4 and we have told that every cycle of length more than 3will be for every cycle of length more than 3 I will have a call joining non-consecutive vertices and hence these vertex one and seven are non-consecutive and hence there is a cord joining it in this cycle 1376.

Similarly we will see that for all other cycles of more than30 of length more than three similarly we will see that for all other cycles of length more than three I will have a cord joining non-consecutive vertices in the cycle and hence this compatibility graph that we have here is a chordal graph now therefore the minimum click partitioning problem for this quarter graph for this compatibility graph can be solved. Now optimally in polynomial time.

(Refer Slide Time: 19:04)



After an understanding of chordal graphs we will now understand what interval graph is simply speaking and interval graph is a graph where the vertices have a one-to-one correspondence with a set of intervals for example let us consider the operation constraints graph here in this scheduled operation constraints graph each vertex can be represented by a continuous single interval defining it start of execution and end of execution.

So each vertex or each operation in this operation constraint graph can be represented by a single Conte continuous interval representing its start of execution to the end of his execution so TI to TI plus di minus 1 this interval is the is the interval for vertex I for each vertex similarly I can define a continuous interval which denotes the lifetime starting from the start of its execution to the end of its execution and hence simply speaking all the operations in this operation constraint graph have a continuous interval and hence the corresponding compatibility graph or the conflict graph is basically an interval graph.

So interval graphs area sub class of chordal graphs if you have an interval graph the property for chordal graphs will always be true so vertices can be put in one-to-one correspondence with a set of intervals vertices are adjacent if corresponding intervals overlap so vertices are adjacent if corresponding intervals overlap so in my interval graph vertices are intervals and there are edges between two vertices if they are intervals overlap now we come to the third class so why have you studied interval graphs we can still have more simpler algorithms for both click partitioning and gruff coloring for interval graphs as we will see later we will study one algorithm for interval graphs for graph coloring of interval graphs now we will look at the third class of graphs a graph becomes a comparability graph.

When it satisfies a transitive orientation property so therefore I need to be able to those are transitive orientation property on my graph so my compatibility graph or my conflict rock can only become a comparability graph if I can impose this transitive orientation property on this graph now if I impose a transitive orientation on an undirected graph it becomes a directed graph so my Compatibility graph or my conflict graph will be a comparability graph when after imposing this direction property on the compatibility graph I will have this in the resulting directed graph g equals to V comma F.

V_I V_J belongs to F and V J VK belongs to F implies that VI VK belongs to F so this direction property this transitive Direction property will be imposed will be impossible on this graph as an example let us see my let us see this compatibility graph that we have here so in this compatibility graph we have we have an edge from one to six and we have another edge from six to seven so let us see this is VI this is VJ and this is VK.

So if I have an orientation if I impose an orientation from V 1 to V6 from VI to VJ I have an orientation from VI to VJ and then I also impose an orientation between VJ and VK I can impose an orientation from V 1 to V 7 so if there is an orientation from Vijay to VI to VJ and there is an edge and oriented edge from VI to VJ there is an oriented edge from V J to VK there will also be an oriented edge from V I to VK this is what this ensures and hence this compatibility graph is a comparability graph.

Because for all the edges I can impose an orientation property like this now on this how can we get a sense of how to obtain this orientation we can assign an orientation with respect to the order of start times in the operation constraints graph now here there is we are putting an orientation from one to six because six starts after one in the operation constraint of six starts after one there is an orientation from six to seven because seven starts after six and hence there is an edge between 12 7 denoting that seven starts after one.

Such an orientation can be can be imposed for all for all edges and hence this compatibility graph is a comparability graph but why are we studying these graphs by studying the chordal graphs and the comparability graphs we can find out whether a graph is inter the rock is an interval graph or not if my conflict rock is a cartel graph and the compatibility graph is a comparability graph then the conflict graph is an interval graph.

So if this conflict graph is a chordal graph here this conflict graph is a chordal graph because there are no cycles of more than three in fact there are no cycles in this graph and hence this conflict graph is a chordal graph we have seen that this compatibility graph is a comparability graph we have already understood this and therefore this conflict graph now becomes a and interval graph right the general theorem is that a graph is an interval graph only if it is Cordell and its complement is compatibility graph here.

We know that the compatibility graph is a complement of the conflict graph and hence if the conflict graph is called the compatibility graph is a comparability graph then the conflict graph becomes an interval graph we come to the end of module 2 of lecture 6.

Centre For Educational Technology
IIT Guwahati
Production

Head CET
Prof. Sunil Khijwania

CET Production Team
Bikask Jyoti Nath
CS Bhaskar Bora
Dibyajyoti Lahkar
Kallal Barua
Kaushik Kr. Sarma

Queen Barman
Rekha Hazarika

CET Administrative Team
Susanta Sarma
Swapan Debnath