

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATHI

NPTEL

NPTEL ONLINE CERTIFICATION COURSE An Initiative of MHRD

VLSI Design, Verification & Test

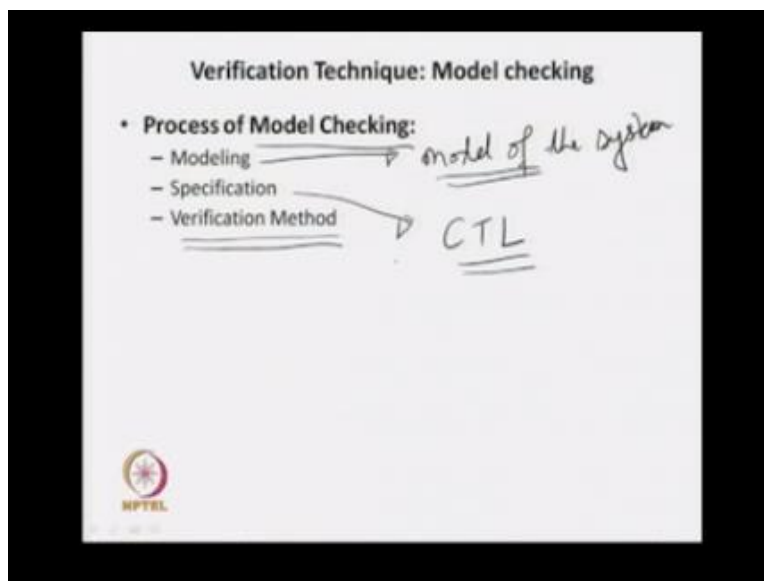
Prof. Jatindra Kr. Deka
Department of CSE
IIT Guwahati

Module V: Verification Techniques

Lecture I: Introduction to Model Checking

So in our last module we have introduced one logic called temporal logic and which will be used to specify the property of our system. Now in this module we are going to look for verification technique and mainly we are going to look for a particular verification technique which is known as your model checking. So in this class I will just briefly introduce what is model checking and what we can do with the help of model checking.

(Refer Slide Time: 00:57)



So I think you might be remembering that in our introduction class okay, we have slightly introduced about that process of property verification and we have slightly introduced about the model checking. Now if you recall back we will find that in the process of model checking we are having basically three components first one is your modeling. So in case of modeling what we have to do, we have to give a model of the systems somehow model of the system.

With some formal we have to define or we have to give the model of our system if we are going to design a new system then we are coming up with design and we are going to represent that design with the help of some model. We will see how we are going to do it, second component is specification or we can say that the property to which satisfy by this particular system or you can say that this is the property that will be satisfied by the model, because we are going to stick out the model of our system.

So in last class we have introduced temporal logic, temporal logic can be treated as a specification likewise and we can use temporal logic to specify our property and we have discussed the special class of temporal logic which is known as your CTL, computational tree logic. So basically we will concentrate on this particular logic CTL and we will see how property can be specified over there.

So one we have the model of the system that we have abstracted the model, we represent our property in CTL or we can give the specification, then we need some methods or some mechanism by which we can check that this property I send it to in the model. So this is the third component which is your verification method. So in this particular course we are basically going to look about the model setting technique which is a verification technique and which is basically property verification technique.

So here in this module we are going to talk about this particular verification method called model checking.

(Refer Slide Time: 03:02)

Model checking

- **Example: Mutual Exclusion**
 - When concurrent processes share a resource (e.g. file or database record), it may be necessary to ensure that they do not have access to it at the same time.
 - Identification of critical section
 - How to model the system
 - What are the specifications

NPTEL

So this model checking we are going to introduce with the help of one example, possibly we will see an example and then we will see how this example will be model or how we are going to give the system on the model. Then we will see what are the properties that need to be satisfied by this particular example and how we are going to write those particular property or specification in CTL.

Once we have the model, once we have the specification in your CTL computational tree logic then we will see that model checking approach or how this can be verified for that particular model. So the example that here we are going to discuss about your mutual exclusion. I think all of you know about this particular problem that it is basically while we are walking with the shared resources.

So when concurrent process shared resource so we can have several resources it maybe a shared memory, we may have a file that file may have shared by different users. On the other hand we may look into the databases also, where the database will be used by different persons or different activities, so in record also you have to see the record of database will be excess from different location.

So in all those cases when we are having the concurrent execution always we have to look for the consistency of our data. If it is a file or if it is a database record or maybe the shared memory of whatever data we have over there it should be consistent, all user should get the similar view of this data. It is not like that I have updated something and that is not reflected for you, then you are not going to get the correct information.

So this is the problem that we have when we are going to use shared resources. So to have a proper consistency of data we can use this particular mutual exclusion principle okay. So here in this particular case we have to say for data criteria that we need to look for it when we are going to model such type of system.

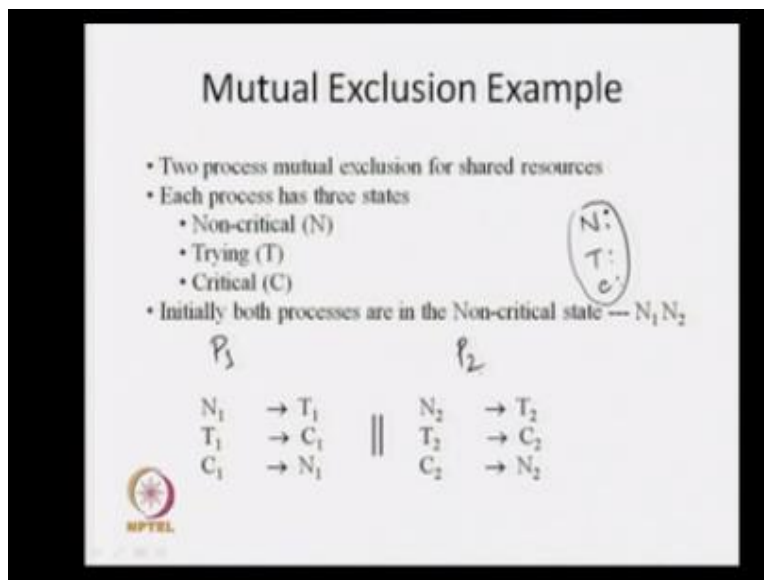
So in this particular case first we have to find out what are the criteria that has to be satisfied by this particular design. So we are talking about that we have to look for those particular criteria that what are the, basically we have to look for that particular portion of code or that particular concurrent processors where they are using this particular shared resources. It maybe your file so we can have a file okay.

So it may be used by several processors for several users, so we can say that process 1, process 2 like that and define process we are having and this N processors are going to use this particular file okay. So all those particular process are having their own code segment. So in a particular code segment I think you can think that it is going to access this particular file. So we have to identify that particular process.

So two such process should not access the file simultaneously that is the restriction we have to say and we say that we have to identify those particular critical section, we say that the file used by this particular code segment we think that is a critical section. So all process may have critical section, so they should not use this particular file at the same time. Now by looking into this criteria we are going to talk about a critical section when you talk about this mutual exclusion then we have to see how to model this particular system.

Now we are going to talk about the concurrent processors, we are going to use some shared resources, so for that we have to come up with a model. And what will the specification of what or the specification that need to be satisfied by this particular model okay. So once we are having the specification we have to come up with the specification, we have to look for the properties, we will come up with the system model and what we are going to say how this, we are going to check whether this specification indeed to in this particular system we said now. So our first that is to come up with the model okay.

(Refer Slide Time: 07:22)



Now you just see that now what is this particular mutual exclusion, we are considering a simple cases where we have said that we are having two processors mutual exclusion, for shared resources. We are considering only two processors these two processors are going to access some shared resources it maybe your shared memory, it may be a file, or it maybe a database record. So for that particular cases what happen each process can be divided into three different step.

We can say that it is your non-critical and trying T and critical C. So non-critical and basically it says that the process is in your non-critical region that means it is not using any shared resources.

So all process can walk in the non-critical regions simultaneously. That state T we are talking about this trying basically we say that now one particular person is trying to enter into non-critical region.

Basically it gives that equation it is time to walk with this particular shared resources and our requirement is like that only one process can use this particular shared resources. One gets the access to the shared resources then we say that the process is in your critical section okay, that means now it is accessing this particular shared resources. So this is critical C, we are saying so we can say that one particular process can be in any one of these particular three state N, T or C.

N is non-critical region, T is it is time to enter into the critical section that means this requesting path is particular shared resources. And critical means now process is executing in the critical region that means it is working with this particular shared resources. Now we are talking about two process mutual exclusion, so we are having two process now say the time process P1 and I am having process P2.

Now process P1 can have in this particular three state and then it was in N1, T1, and C1 similarly process two can be in three states it is either in N2, T2 and C2. Now what are the condition you should see that if it is in the N1 that means it is in your non-critical section now at some point of time it may go to the time state, that means now it is trying to enter into the critical section. Once it is entering into the trying state that means now it is trying to access the shared resources.

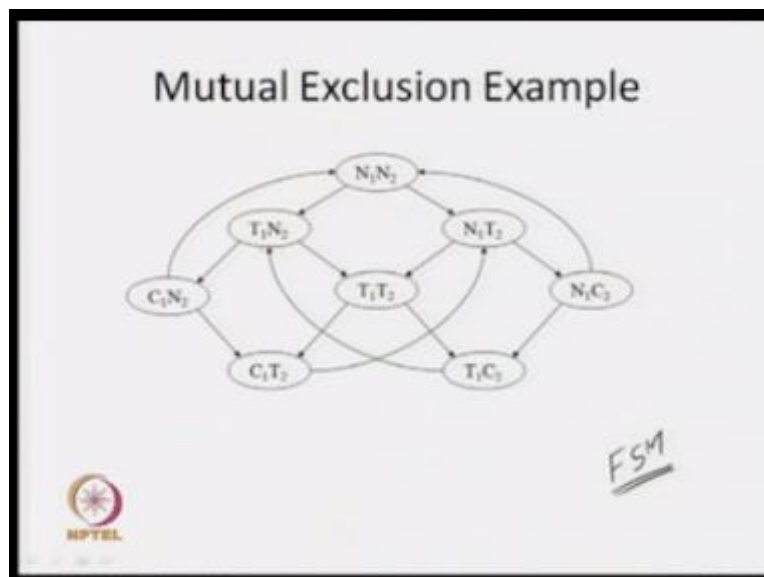
Now if it gains the shared resources then what happens it will go into the critical section that means it is going to use this particular shared resources explicitly it is going to use this. Once it zobbies a word and what will happen now it will come out from critical section to non-critical section that means it is going to do the other normal execution. So this is the transition about process P1.

Similarly for process P₂ also we have this particular similar transitions so an 22T2 T22 TC2 and CT C22 and 2 so these are possible transition that we have one process P₁ and process P₂ now we are going to see these things as an as one system and for that path we are going to do that

behavior of these two process P_1 and P_2 will be composed together so now we said this the failure composition.

Now when we are going to get a composition of this process P_1 and P_2 then we are going to get the total behavior of our system now how now we will see what this total behavior is look like.

(Refer Slide Time: 10:48)



Now eventually we can come up with such type of finite state machine why I am taking out already I have said that this is your same sort of finite state machine we are having finite number of states and we are having transition among them now what are the states you just see that first I am talking about say this is the given as your $N_1 N_2$ what does it means it says that the process P_1 is your non critical section process N_2 is in your non critical section so both are executing they doing their own job but none of them are using the critical section.

Now when the state is your N_1 and N_2 then what will happen at some point of time process P_1 may look other cheer discos so it will go from N_1 to T_1 do it is having a transition from N_1 to T_1 so we are having a next state $N T_1$ and N_2 it is an process P_2 is an non critical section and process P_1 is your time section at the same time it happened that now process P_2 may look further

particular see at resources so we may have another transition so it is from N_1 and N_2 to N_1 to P_2 okay.

So you just see that initially both are in non critical regions now either P_1 try to enter to the critical section so it will then the states we are going to get is your $N_1 P_1 N_2$ and one process P_2 wants to enter into the critical section then we are going to get the state $N_1 P_2$ now when it is in your T_1 and N_2 you just see either now process P_1 is going to get the see at resources so T_1 will enter into the critical section and N_2 remain in the process P_2 remains in the enter section it is in the so non critical section but it may also happen that now once is trying that process P_1 is trying similarly process P_2 may also try it try to enter into the critical section so the state we are going to have is T_1 and T_2 .

Now after some point of time that process your distance of the call P_1 will come up for the critical section so we are going to have this particular N_1 and N_2 state and now one similarly this path is symmetric to the other path that it is from P_2 it is going to C_2 then it will come out and it will go from N_1 and N_2 now one both are entering in trying to enter in to the critical section so one will go to say process P_1 will go into the detail section so it will follow this particular path and process P_2 will go into the critical section then it will follow this particular path and from there it will come to this particular P_1 state.

So you just see that with this particular state transition machine we have basically given the behavior of our system so what we have said mutual exclusion.


(Refer Slide Time: 13:40)

Mutual Exclusion Example

- Two process mutual exclusion for shared resources
- Each process has three states
 - Non-critical (N)
 - Trying (T)
 - Critical (C)
- Initially both processes are in the Non-critical state --- $N_1 N_2$

P_1 P_2

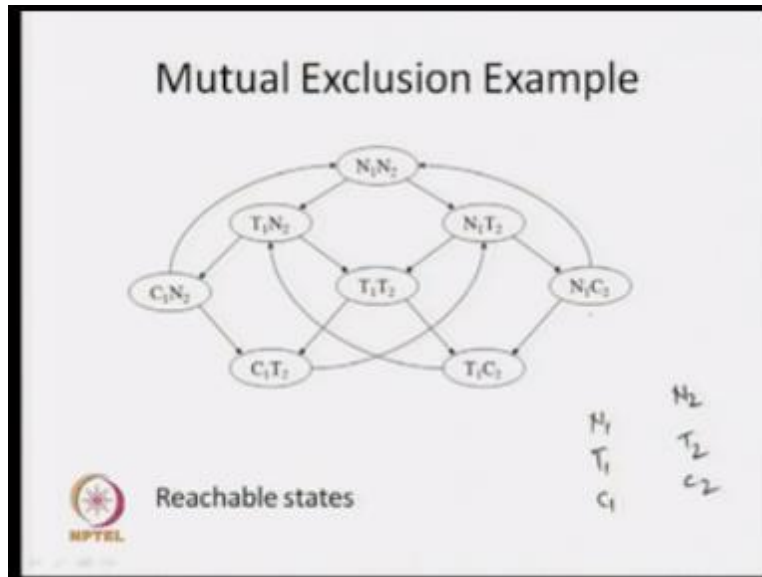
N_1	\rightarrow	T_1		N_2	\rightarrow	T_2
T_1	\rightarrow	C_1		T_2	\rightarrow	C_2
C_1	\rightarrow	N_1		C_2	\rightarrow	N_2



A handwritten circle around the states N, T, and C in the list, with an arrow pointing to the initial state $N_1 N_2$.

We say that we are having two processes P_1 and P_2 these are the possible states and these are possible transition now when we are looking into them as a whole then we are just getting these particular transition behavior now you just see that we are getting this particular transition behavior.

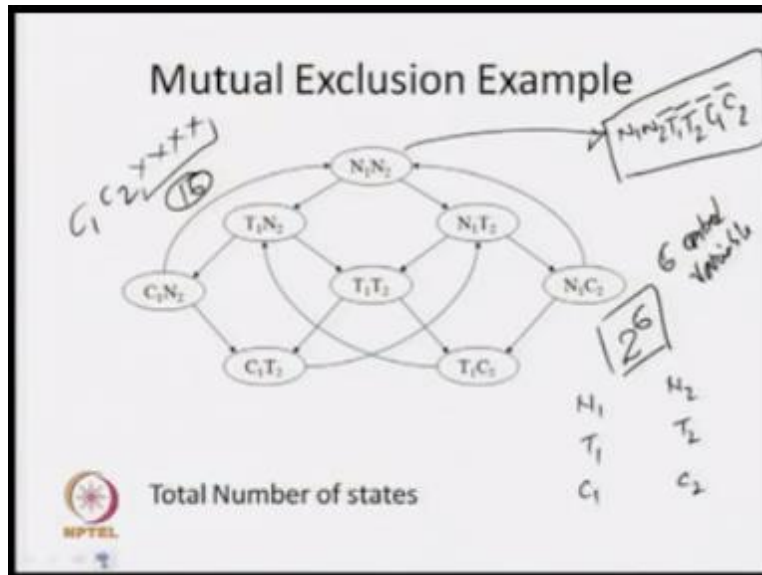
(Refer Slide Time: 13:52)



Now you just see that we are getting as transition behavior and this transition system is going to detect the behavior of our system okay now here we are talking about say some state variable we are saying that we can have for process P₁ we are having N₁ P₁ and C₁ for process P₁ we are N₂ P₂ and C₂ out of that if you look into this particular behavior we say that we are getting 1, 2, 3, 4, 5, 6, 7, 8 total 8 different states and we are having some transition.

So these are the states that we are going to talk about these are reachable state of my system so if my system is work going to work correctly or work perfectly these are the different possibilities that we are having so these are the different states that my system can go and we are going to talk about the these are the reachable states if we are talking about a reachable state then we may have some other states which may be non reachable state.

(Refer Slide Time: 14:59)



So for this particular example you can think that what are the total number of states that we may have now you just see that already I have said that we are having $N_1 P_1$ and C_1 for process P_1 and $N_2 P_2 C_2$ now when I talk that in this particular state I am talking about that this is your $N_1 N_2$ that means both are in your non critical section so these are some state variable we can say that these are true basically N_1 is true and N_2 is true and what are the states of other 4 variables.

We can say that this is the not trying to enter in the critical section so it is T_1 bar there process P_2 is not trying to enter into a critical section so it is T_2 bar similarly process P_1 in the critical section so it is C_1 bar and similarly process P_2 is also not in the critical section so it is your C_2 bar so you just see that with this particular combination basically we are representing this particular states.

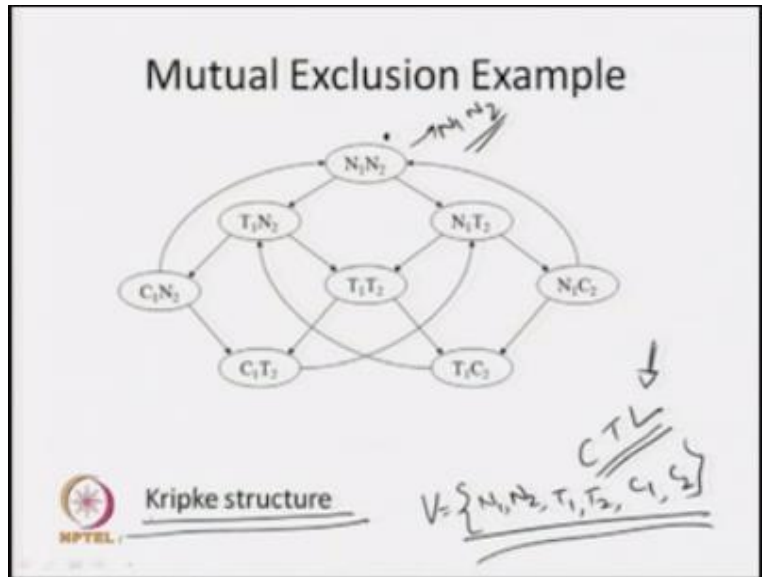
Okay now in this particular case so it is basically N_1 and N_2 are true that means both the process are in a non critical section so similarly T_1 is false and T_2 false they are not trying to enter into a critical section and C_1 and C_2 are also false they are not in the critical section so like that for all this particular states we can have the behavior of this other variables now in this particular case we having written 6 control variable.

So if we are having 6 control variable what is the different possible combination of this particular control variables we will see that we will get all together 2^6 different states because if we are n control variables then we are going to get 2^6 different states so that means my total state space is your 2^6 okay which is at a 64 that means I am going to get 64 different states this is the total number of states of this particular system but out of that what will happen we have seen that only 8 are reachable.

Other are not reachable because for a proper system both cannot be in the critical region that means C_1 and C_2 may not should not be true simultaneously so it is a C_1 and C_2 then we having other 4 variables T_1 T_2 and N_1 N_2 so see this particular 4 variables now have either true and false so that means we can get with 4 variable we are going to get 16 different states that means this 16 different states are not reachable it is not the proper behavior of our system and we are going to say that these are the not reachable states.

Okay like that some other states will also go out form this particular model and eventually we are getting only these 3 possible states okay so these are the reachable state you just see that when we are trying to model our system depending on the input variables are the control variables you may have bigger states space but all states may not be relevant for us and we are going to basically concentrate on the reachable states so these are the reachable states.

(Refer Slide Time: 18:24)



Okay now you just see that when we discuss about your temporal logic or CTL computational tree logic we have mentioned that the meaning of this particular CTL is defined over a model okay and what is a model if you recall back you will find that this is nothing but some sort of finite state system we are having finite number of states we have the transition so this is basically behavior above finite state system but along with that we are having one additional function which is known as your labeling function.

Each state will be labeled with the atomic proposition if the true atomic proposition is true in a particular state we are going to label this particular state with the help of this particular atomic proposition now when we are coming up with this particular model for mutual exclusion here we will find that the set of atomic propositions V will be $\{N_1, N_2, T_1, T_2, C_1, C_2\}$ so this is the set of atomic propositions that we have in this particular model and you will just see that in this particular state we are marking with N_1 and N_2 .

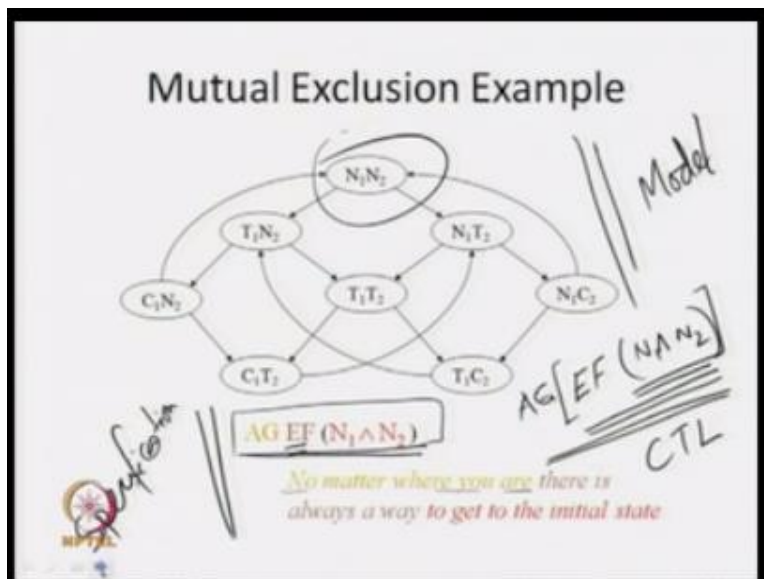
So what does it mean that means this particular atomic proposition N_1 is true over here the atomic proposition N_2 is false over here and other atomic propositions T_1, T_2, C_1, C_2 are false at the particular states now you just see that when I am coming up with this particular model this particular

model now very much similar to our Kripke structure that is used for our in CTL to define a meaning of our CTL competition three logic formula.

So now eventually you have come up with this particular Kripke Structure, so for model checking we need a model we need a specification, specifications are nothing but the properties that should be satisfied by this particular system. And we need some techniques by which we are going to check whether this particular properties are two in or model or no.

So now the first part we need some model we have come up with this model of this particular examples of mutual exclusion and eventually we have found that the way we are representing it, it is going to give us the Kripke structure that is needed to define the meaning of much CTL formula, okay. So in last module we have talked about this particular CTL what is a syntax of CTL and what are the symmetric of CTL, now we have come up with the model.

(Refer Slide Time: 20:59)



Now user we know when we are coming to this particular models we have come up this particular model and this is very much similar to our Kripke structure, now I think now I am I think over here is a see that $AG [EF (N_1 \wedge N_2)]$ now if you look into this particular expression I

think you can recall that this is very much similar to our CTL formula and in that this is a CTL formula okay because you just see that $N1$ and $N2$ these are two atomic proposition and they are connected by this particular conjunction so $N1$ and $N2$ is a CTL formula now this is EF, what is EF there exist a part in future.

So there exist a part in future $N1$ and $N2$ so I am going to get this is also CTL formula since this is the CTL formula $EF(N1 \wedge N2)$ will be the CTL formula and before that I am writing AC that means in all part globally there exist a part in future $N1$ and $N2$ that means I can for create it I can give this particular backups so you see that since this is the CTL formula so eventually this is the CTL formula.

So now I am writing a CTL formula now what this CTL formula means you just see that what we can see in sentence no matter where you are here is always a way to get to the initial step so no matter where you are that means in a system wherever you are it hardly makes any difference no matters there is always a way we are going to get a part to get the initial step here we have disgusting thing that this is my initial step okay.

So this is why I am said that in initial step I may be $N1$ and $N2$ must be true so no matter where you are there is always a way to get to the initial state now this is the property that need to be satisfied this particular model now you just see that I am having model say this is the model of my system as well as prism, this is the specification out of property, one simple specification that I am mentioning over here.

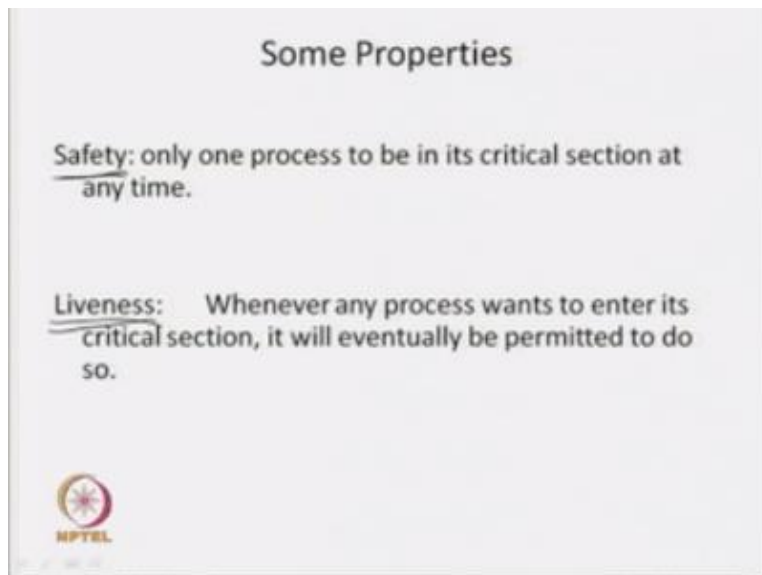
So specify in our property now I need some mechanism to check whether this property or this specification is true in this model or not, okay and this particular mechanism or technique is known as your model second, this model second is that kind of verification technique it is a kind of verification techniques and in this class we are going to or in this course we are going to talk about this particular models again.

So from model second what is our requirement we need a model of the systems somehow we have come up with the model which is very much similar to the Kripke structure we have to

represent our property or specification in CTL Formula then we are going to use model checker which is basically known as a CTL Formula model checker in this particular case since we writing the specification in your CTL Formula.

So we are going to look for CTL model checker and CTL model checker is going to check whether this particular specification is true in this particular model or not, so this is the way that we are going to look in verification technique and in this course we are going to talk about model checking.

(Refer Slide Time: 24:37)



And particularly in model checking above CTL Formula because we are going to write our formula in CTL now when we are talking about this particular we see a exclusion problem of our shared resources, now it should satisfy some over properties we are having some recommend it should full fill those particular requirement, first we have to identify what are those particular requirement.

So one requirement we are talking about that safety requirement what is this safety requirement it says that only one process to be in its critical section at any time, this is our basic requirement

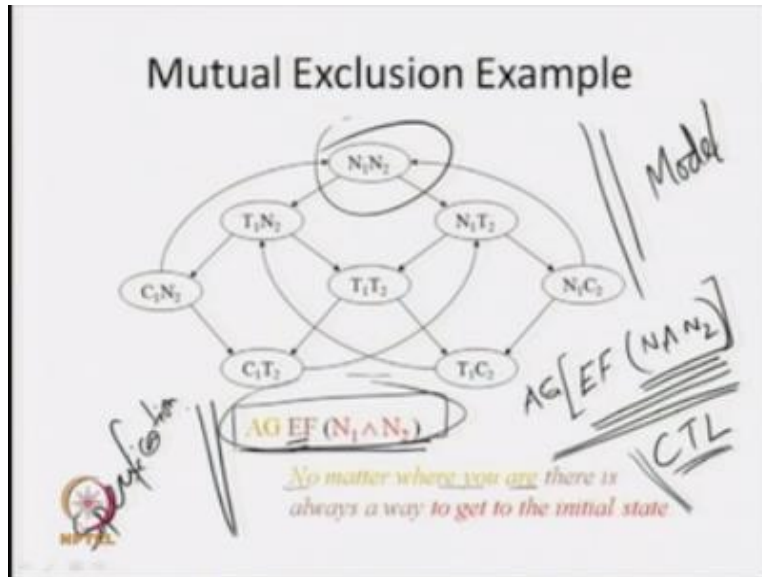
and say that this is the safety requirement so at any point of time only one process will be in its critical section, if two persons are going to remain in the critical section at the same time so both of them are going to manipulate the data's.

And the manipulation of one process may not be reflected in the other because it will be a word written by the other process, so this is basically safety property only one process to be in its critical section at any time, now second property we are talking about the liveness, what we are talking about liveness, whenever any process wants to enter its critical section it will eventually be permitted to do so.

So we are talking about the liveness that means all processes are having equal right to enter into the critical section, so whenever a process wants to enter into critical section that means it wants to use the shared resources eventually it should get the permission to use that particular shared resource, so this is about liveness if someone is waiting and waiting for long enough time then can set at that process going to safety zone so it is not desirable.

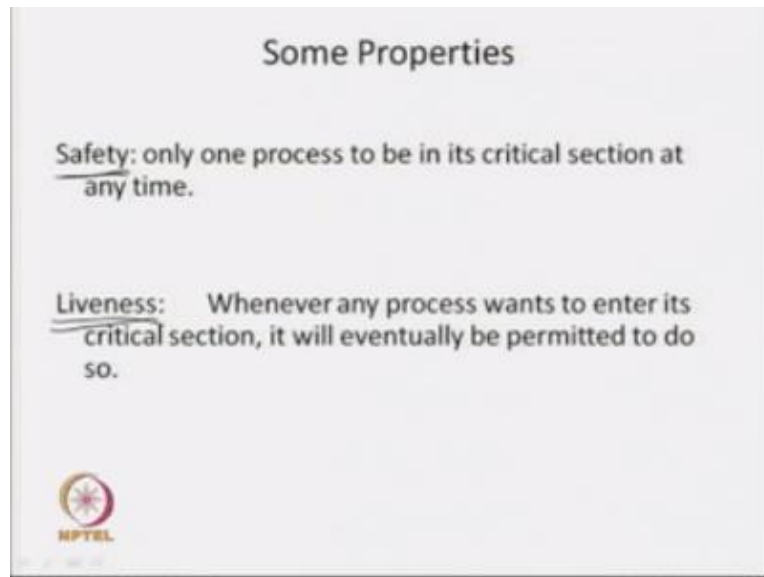
So the second property is your liveness now say we know that these are the two requirements that we are having, now whether if I am having a whether if I said that whether these two properties are satisfied this particular model or not, I think then we will go or we will move with the some other domain when we have to look for the natural language is passing because we are representing our requirement in natural language. So we have to use some formulas in or use some common matter to represent such type of properties.

(Refer Slide Time: 26:52)



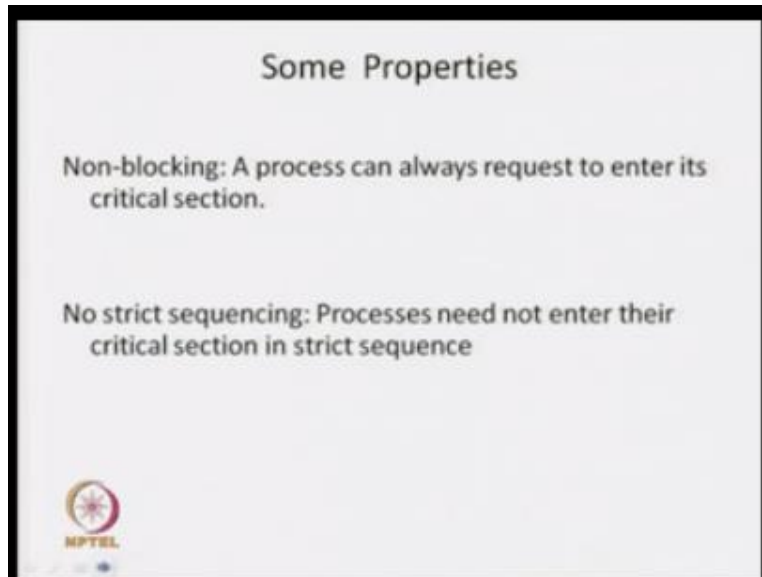
Say like that in my previous example I am saying that no matter of where you are here is always a way to get to the initial step so this is some sort of requirement I am represented this requirement with the help of this particular CTL Formula so some because this CTL Formula have been pre-defined meaning and we have to use some pre-defined syntax to write such type of formula, since meaning is obvious. So now we are going to look for the truth of this particular formula in this particular model.

(Refer Slide Time: 27:23)



So like that these are two properties that must satisfy by my mutual exclusion protocol somehow we have two represents these things you know in some formula and since we know about CTL competition physiology we will see how these two will be represented in CTL competition pre logic, okay. Now another properties that we are having that non blocking A process can always requires to enters critical section.

(Refer Slide Time: 27:50)



So this is non blocking property we should not have some critical for some time we are going to block one particular process that process will not be allowed into critical section that we said that this is some blocking section so we say that our we say let us some protocol mass satisfied is particular non blocking property also the air process can always request to enter it critical section.

So at any point of time in any moment any persistent you are request to enter into critical section you should not block any blocking criteria and another criteria or another properties that must satisfy by this spacial expression protocol this is no strict sequence here we should not give any sequence any strict sequence to that particular process it is first sequence we cannot say that process p1 will go into the critical section then I will allow process P2.

Then I will allow process P3 then again I will allow process P1 to enter into critical section then P2 like that we should not put any restriction in this particular case without any restriction without any sequencing they will be allowed to enter into critical section, it may happen at P 1 as entre into critical section after sometimes again P1 may allow to enter into critical section we will not wait for the first P2 must go then only again P1 will be allow.

So such type restriction we are not going to put so this is talk about no strict sequence, so basically we said that our basic exclusion your protocol must satisfy this particular four properties, now already I have mentioned that now to be understand it properly or when we have to specify this thing and here in this particular course we are going to for the CTL representation of those particular properties. Because we are going to for CTL model checking.

(Refer Slide Time: 29:31)

Some CTL Properties

Safety: only one process to be in its critical section at any time.

$$\text{AG } \neg (c_1 \wedge c_2)$$
 $\text{AG } \neg (c_1 \wedge c_2)$

Liveness: Whenever any process wants to enter its critical section, it will eventually be permitted to do so.

$$\text{AG}(t_1 \rightarrow \text{AF}c_2) \quad \text{AG}(t_2 \rightarrow \text{AF}c_2) \quad \text{AG}(t_1 \rightarrow \text{AF}c_1)$$

MPTEL

Now the first property is the safety property you can say that only one process is to be allowed in its critical section so we can write this particular CTL Formula like that in all part globally $\neg(C1 \wedge C2)$ so if C1 and C2 if it is true then what will happen both the process in a critical section so I am going to eventually so \neg of this must be true, okay and where it must be true, in all part globally that means in the enter system in all steps in along all part is must be true not of c1.

And c2 that is why you all I think this property has in all part globally not of c1 and c2. Now you just see that this is a CTL formula because c1 and c2 is a CTL formula because c1, c2 is our atomic proposition, negotiation of CTL formula is also a CTL formula and AG is a temporal CTL operator globally in all parts so this is a CTL property.

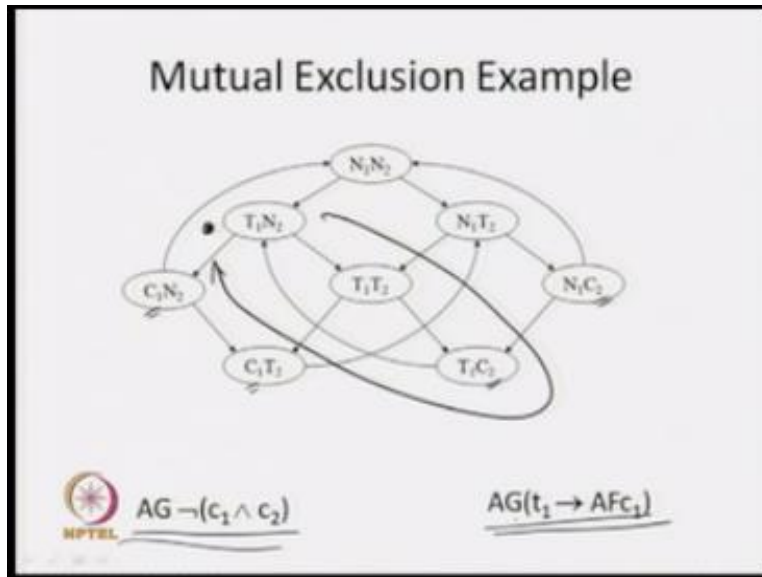
So we are writing this safety property in CTL formula, similar liveness we are saying that whenever any process wants to enter into critical session it must be allow to enter into the critical session, so we can say that $t1$ basically it says that the process $p1$ is time to enter into the critical session then if it is true then what will happen in all part in peruse $c1$ so it process $p1$ is trying to enter the critical session then from that particular point.

Wherever you go in all part in peruse that process must go into the critical session so that is why I am saying that once to enter its critical session it will eventually be permitted to enter so, to so. So that is why I am saying that if $t1$ is true then it implies that in all part in peruse we should get a step where $c1$ is true and this must be true in all steps so we are saying that in all part globally this is true so this is the way that we are going to write.

Now you just see that these properties that I am writing $AG\ t1$ implies $AFc1$ this property is basically related to the process $p1$ because we are having now two process one process $p1$ and process $p2$. Now similarly we can write a formula for process $p2$ so it will look like or it will come up like that AG it process $t2$ is time to enter the critical session then in all part in peruse $c2$ must be true so this is the property related to your process $t2$ so process $t1$ having this particular property process $p2$ is having this particular property.

But in case of safety that AG not of $c1$ and $c2$ this is for the entire system so this is for both process $p1$ and $p2$, okay so now just see that we are having safety property and we are having liveness property these two property have represented that with the help of CTL formula.

(Refer Slide Time: 32:40)



So these are the two formulas for safety and this is your liveness now we have to check these two property must be true in this particular system. Initialization just I am going to look for this one say $AG \neg (c_1 \wedge c_2)$ so if you see that we are having eight possible step in eight possible step both c_1 and c_2 is not true that means in all step this is true because here in this particular four steps c_1 and c_2 is not true but here c_1 is true and c_2 is false.

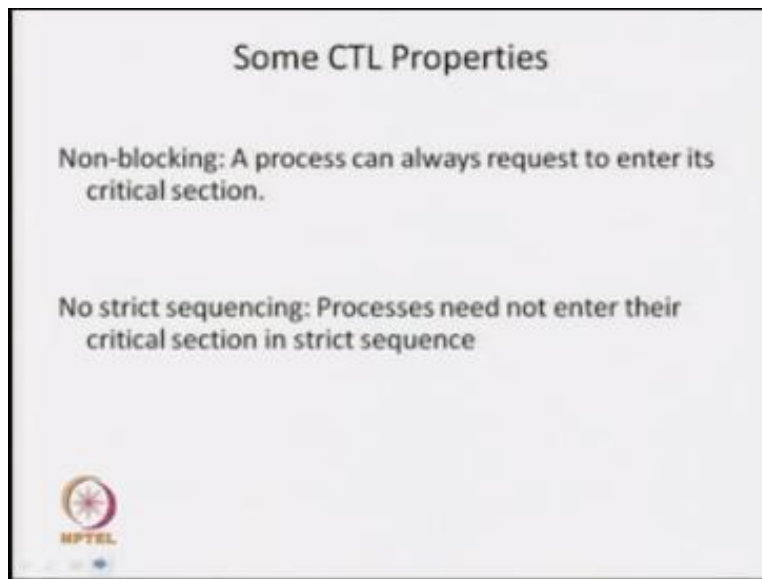
And in this particular case c_2 is true, c_2 is true but c_1 is false, so this $AG \neg (c_1 \wedge c_2)$ is true in this particular case. Now second property what we are talking about, whenever any process wants to enter its critical session it will eventually be permitted to do so, so if it is trying to enter into the critical session in all part in peruse you should get c_1 , so all part globally te implies AFc_1 , so in all part globally c_1 .

But if you look into it here I will be having slight problem I will just mention it say, if you consider that I am in this particular step, a process p_1 is trying to enter into the critical session if I pull out this particular part then it is entering the critical session, but it is having another part also now you consider this particular part say from $t_1 n_2$ it is going to $t_1 t_2$ then it is going for $t_1 c_2$, now process p_2 is entering into the decision and it is going back.

Now you consider this particular part, okay now if you consider this particular part then what will happen will find that in this particular part we are not going to get $c1$ that means process $p1$ is not entering into the critical session, so whatever I will say that these particular formulas false that means the model that I have come over here is not setting, it is saying the liveness property because first one is safety property we have satisfy, safety property is has been satisfied by this one.

But when I am coming to this particular liveness property we have seen that it is not satisfying this particular liveness property. Because we are having one particular part it will remain in this particular part and always $c2$ is entering into this critical session, okay.

(Refer Slide Time: 35:10)




Now similarly we have talked about the non-blocking and we are talking about non-strict sequencing so we should not block any process to enter into the critical session and we should not put any strict sequence that is process must enter in this critical session in some predefined or strict sequence so we should not put such type of criteria.

(Refer Slide Time: 35:36)

Some CTL Properties

Non-blocking: A process can always request to enter its critical section.
$$\underline{\underline{AG(n_1 \rightarrow \underline{\underline{EXT_1}})}}}$$

No strict sequencing: Processes need not enter their critical section in strict sequence
$$\underline{\underline{EF(c_1 \wedge E[c_1 U (\neg c_1 \wedge E[\neg c_2 U c_2])])}}}$$



So these two properties also we can represent with the help of CTL formula, so it says that it is always request to enter its critical session so it a process p1 is in your say non-critical session so their exist a part the next that it is t1, okay that means it process p1 is in your non-critical session from that particular step we should get at least one part where it is time to enter into the critical session that means we are not blocking wherever you are if you are in the non-critical session we are having you are having prohibition to try to enter into the critical session.

And that strict, non-strict non sequencing no strict sequencing we can represent this particular property with the help of this particular CTL formula, what it says that since it is non we do not put any strict sequence that means it is basically says that if it is entering into the critical session say it is in a critical session it may happen that after coming out from critical session it will again allow to enter into this particular critical session that is why we are saying that if it is c1 is true and their exist the part c1 will remain to until this particular property satisfied not of c1.

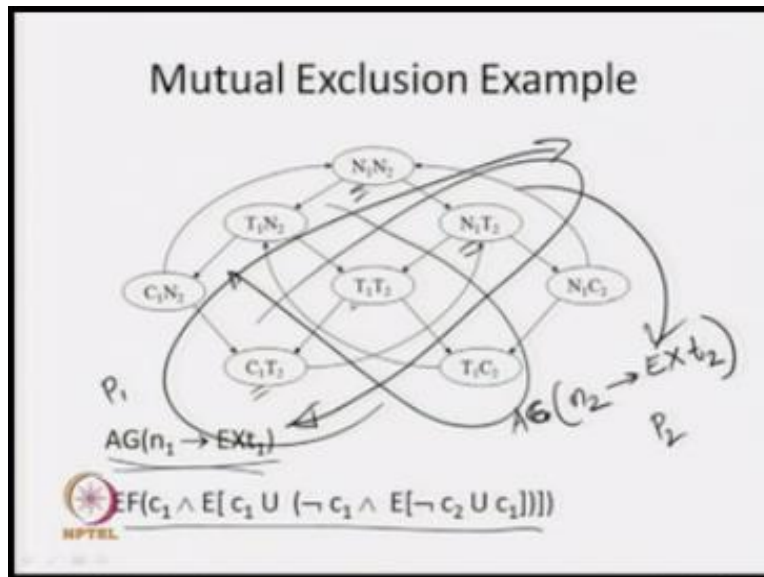
That means c1 go to, it will become negotiation along with that c2 will remain negotiation so c2 will remain false until again c1 becomes false, so you just see that. Here we are saying that we are in the c1 we have entered say you can say that in this particular we are in c1 I can come out

from this particular point we will go to that particular say not of c1 that means now c1 is false I am coming out from this particular critical session.

Now what will happen you should get a part such like that I can again my system will go to this particular c1 but in none of the state c2 will be must be true that means c2 will remain first that means c2 is not entering into the critical session so this is the way that I am representing this particular behavior so we are saying that not of c2 remains true until again c1 becomes true, so you just see that again this is a CTL formula because it is satisfy or it is satisfying the requirement of our or recommend our syntax about CTL expression, okay you can checked it.

Because we are said that we are having some temporal operator until only we are using over here and this particular peruse also we are using and we are using this particular part quantifier E and F, okay. So this is the way that we are represented this, now we have check whether this property is a true or not.

(Refer Slide Time: 38:19)



Now about this particular things AG N1 Ext1 we will find that whenever n1 is there in next step I am having said t1, okay I am having n1 then you can say I am getting t1 so if it is n1 that means

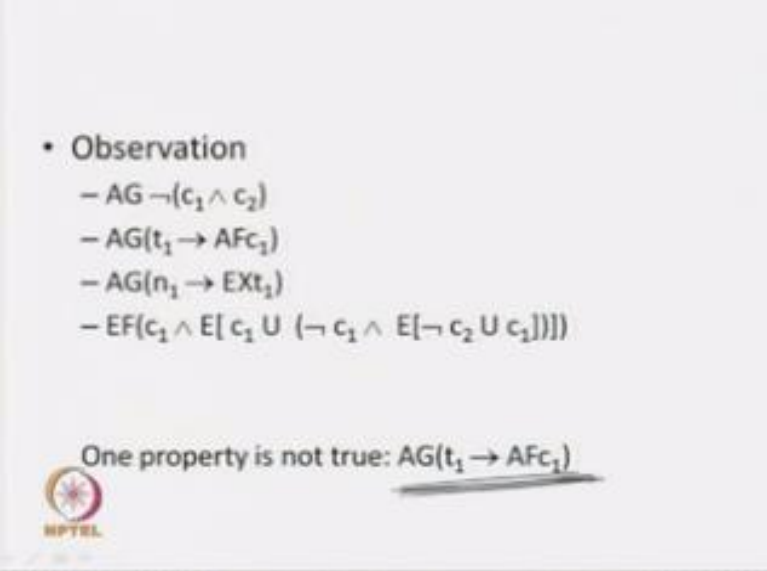
it is in a non-critical session their exist the part in next step it is your t1, okay so it is getting a state where from this time to enter into the critical session, so this behavior is related to your process p1 similarly for process p2 also I can write the behavior of property in all part globally if n2 is true then it says that their exist the part in next state t2.

So this is the property corresponds to the process p2 and this is the property corresponds to the process p1. Similarly here whatever I am saying wherever n1 is true I am basically concern about this particular state and we are going to say that in next step t1 is true, okay because if n1 is false basically this implication is always going to give me true by look true, okay I will see this and this is again the property that we are having which is related to process p1 similarly for process p2 what happens we can say that this c1 will be replaced by c2 and c2 will be replaced by c1 then it will now property will be relevant for your process p2.

Again just I am giving the intuitive idea about it said this we can say that this property is again true over here because we are going to get one particular part where this particular property is true, because this is you can related with the liveness property because we have said that in this particular part c1 is not getting so every then c2 is entering so similarly I am can look into this particular part now, okay in this particular part what will happen always process p1 is going into this critical session at no point of time process p2 is going into the critical session you just see that, that means in this particular part I am allowing to enter process p1 to in its critical session, okay.

So in between process p2 is not getting any sense that means we are not following any sequence or what so c1 can enter into the critical session for more number of times and eventually we can say that after some times from here it will come out and it will go to this particular process p2 will go into this process critical session. Now you just see that here we are talking about this particular visual expression problem.


(Refer Slide Time: 41:06)



• Observation

- $AG \neg(c_1 \wedge c_2)$
- $AG(t_1 \rightarrow AFc_1)$
- $AG(n_1 \rightarrow EXt_1)$
- $EF(c_1 \wedge E[c_1 U (\neg c_1 \wedge E[\neg c_2 U c_1])])$

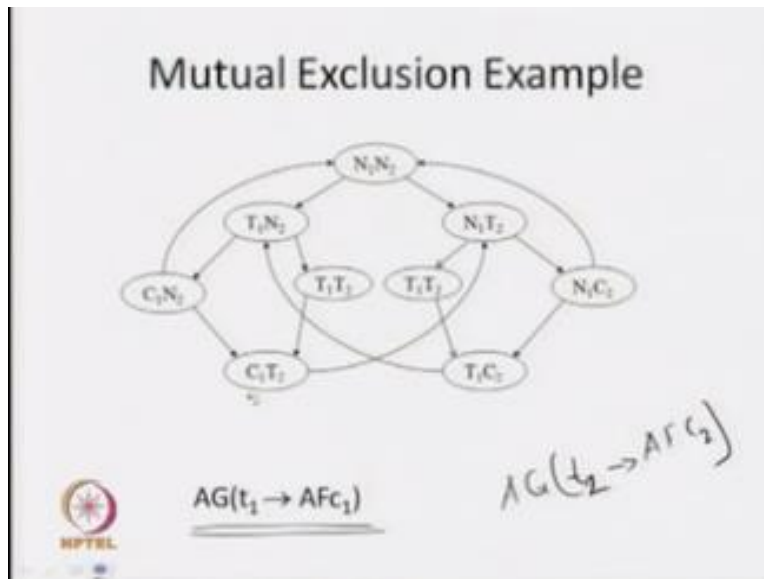
One property is not true: $AG(t_1 \rightarrow AFc_1)$

 NPTEL

And we have come up with the model and along with that we have seen in particular particular specification this is safety aliveness then no stricken and non blocking and out of this particular for formula what happens we have found that this property is not true in my model now say I am trying to design my this particular critical section or exist a critical section of basically may by missal exclusion I have design it now I have seen that it should satisfy four requirements for properties we are come up with four properties out of that four properties one is not it is not satisfying already have seen.

Now in that particular case now what I have to do as a designer I should re look my design and try to modified my design in such a way that it is going to satisfy this property this is the way that we are going to design our system so since this property is not true then I will revisit my design and I will try to check where is the problem and I will try to picks up that particular one. Now after looking to this what will happen now I can think something like this?

(Refer Slide Time: 42:19)



So this is the model that I have come up and I have seen that this particular property is not true over here okay because I have already seen that this is the part that we have now problems happens in this particular part what will happen process p1 is not getting any sans to and turn in to this particular critical section C1. Similarly for process p2 I have rind $AG t_2$ in plus C2 since this for what is not true similarly this property is also not true because we are going to get this particular part over here.

Where process p2 is not getting any sans to and try to get this also that means you just see that I am having problem in this particular portion in my design I am having some problem over here it may not be problem with the entire system, so I will concentrate in this particular part and I try to modify my design so in that particular case what will happen someone help to break this particular two loops okay.

Now what will happen in this particular case now I can break this particular two loops and I can come up with this particular design okay so this is the way that we are going to these are our system and we are try to remove some errors of fall that we may have in our design okay. Now ion this particular case now you just see that what will happen now earlier we have 8 steps now

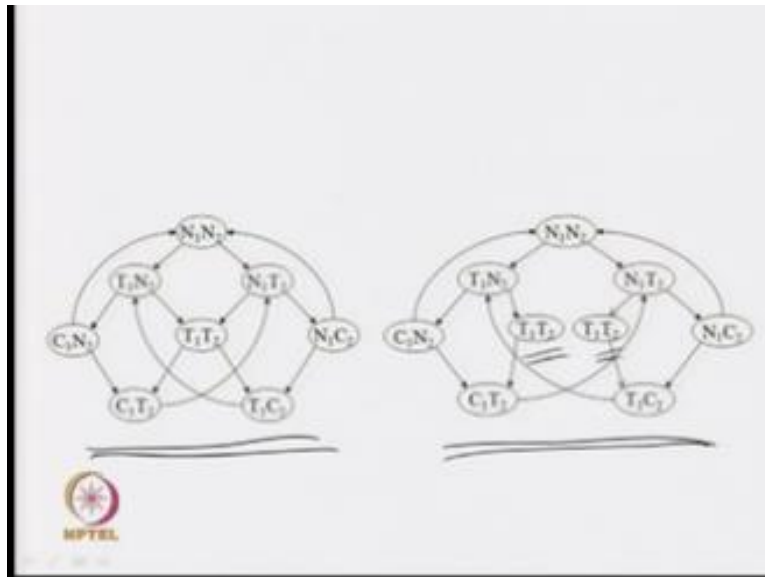
we are getting nine step this particular step has been broken in to two particular step t1 and t2 has been broken in to two different step okay.

So now in this particular case what will happen you just see that now when I am having that t1 it is I am try to entering to the t1 then it is entering in to your c1 now after that I am coming back to that particular state where n1 and t2 okay. Now from here either it will go to t1 and t2 or can get n1 or c2 that means it is already enter in to this particular critical section okay. So this is the way that we can have a formula what we can say that t1 and c2 now from here we are going to t1 and n2 and eventually where I am coming to this particular c1 either this direction or from this direction.

You just see now by after breaking this particular two loops I am getting coming up with the design in this design you will find that this two this formula is true over here okay similarly that is other formula $AG(t_2 \text{ implies } AFC_2)$ will also be 2 that means that plainness property will be satisfied over here okay. So this is the way that we are going to say what happen after applying this particular verification again or try to check for the truth of this particular property I will found that there is some problem.

So now we visit our design and w have modified it and we have seen that now this particular design is going to satisfy my liveness property now when it is satisfying my liveness property then we have to see whether it is not spoiling my other three properties okay in that we can say that other three properties also remain in take over here okay you can which check it over here that other three properties are true over here.

(Refer Slide Time: 45:51)



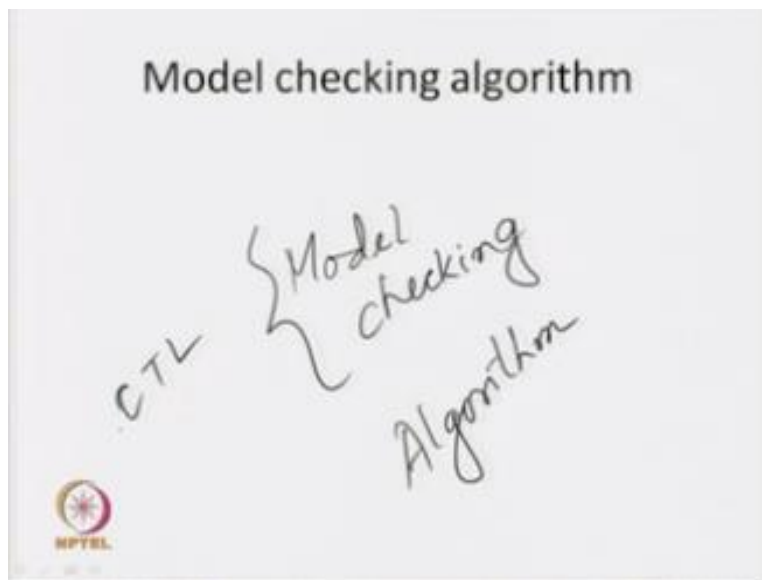
Now basically you just see that we have starting our design we have come up with the model m and after looking for the properties which try to check whether this properties are true over here or not and we have found that it is not satisfying some of the property we are revisit our model and we have to come up with this particular model second model and in this particular second model we have seen that at least it is satisfying all those particular four specification or four properties that we have okay.

Now say yes for the norms of the keep gust structure we having slight problem over here that two step cannot be mark the same level some of they have to be different now here I am not talking in to this particular internal details but we can say that instead of this particular six step that I am having of the process that n_1 p_1 c_1 and n_2 t_2 and c_2 we may have some other internal signal also in the controller so these two steps will be distinguish by some of this particular control signal.

Okay in that up one the one we are look in to it but you can think that they are different because you may have some other conclude signals which are not relevant with respect to our properties so we are not showing the marking of those particular things. Now what we need next now we

are having that model we are having the property CTL properties we are representing our specification with the help of CTL property, now we need a mechanism by which we are going to check those particular properties true ion this particular model or not.

(Refer Slide Time: 47:28)



So for that we are going look for a particular model which known as your model checking okay so we are going for this particular model checking and basically we are going to talk about CTL model checking because our specification will be represented by CTL formula and we are going to look for an Algorithm team by which we are going to check whether a particular CTL formula is true in our model or not.

So we are going to look for model checking algorithms now what is this model checking algorithm how we can visualize it.


(Refer Slide Time: 48:06)

$M = (S, \delta, \lambda)$
 M, ϕ
 M, s_0
 M, s_0, ϕ

Model Checking Algorithm

Given the model M , the CTL formula ϕ and a state s_0 of S as input

Model checking algorithm generates answer 'yes' ($M, s_0 \models \phi$ holds), or 'no' ($M, s_0 \not\models \phi$ does not hold).

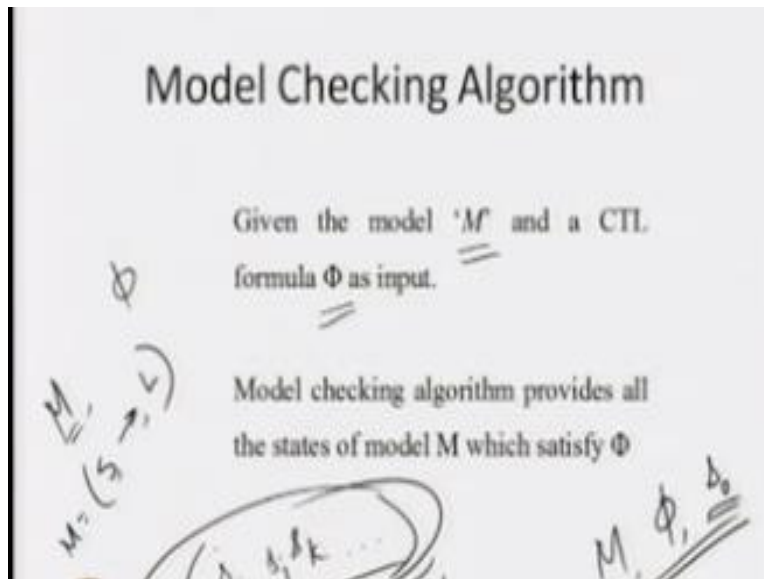


You said that given a model m so we are having a model m the CTL formula ϕ so we are having a model m we are having a CTL formula ϕ and along with that we are having a step s of S that things we are going to look for any state s_0 which belong to the state of step of s of this particular model. Because model m is basically defined by your state of step transition function and leveling function okay.

So what is the model checking algorithm so model checking algorithm generate and say yes it will say yes if π s_0 model ϕ on that means you can say that ϕ holds in your $m s_0$ of ϕ is true in your s_0 so that model checking algorithm generate and said yes if π is true in this particular step s_0 or it generate and say no if ϕ does not holds over there so m as model V does not holds that means ϕ is not true in s_0 .

So this is the way we can look in to the model checking algorithm that we are giving a model m we are giving a CTL formula ϕ and one particular step and we are going to check whether this particular CTL formula is true in this particular state s_0 of m or not. So it is going to say wither yes or no.

(Refer Slide Time: 49:31)



The model checking algorithm maybe put in the another way also it says that we are giving model m and a CTL formula ϕ so we are giving a model m and a CTL formula ϕ so model m I can saying that it is having state of step transition relation and leveling function now model checking algorithm we can now view it like that model checking algorithm provides all the states of model m which satisfy ϕ .

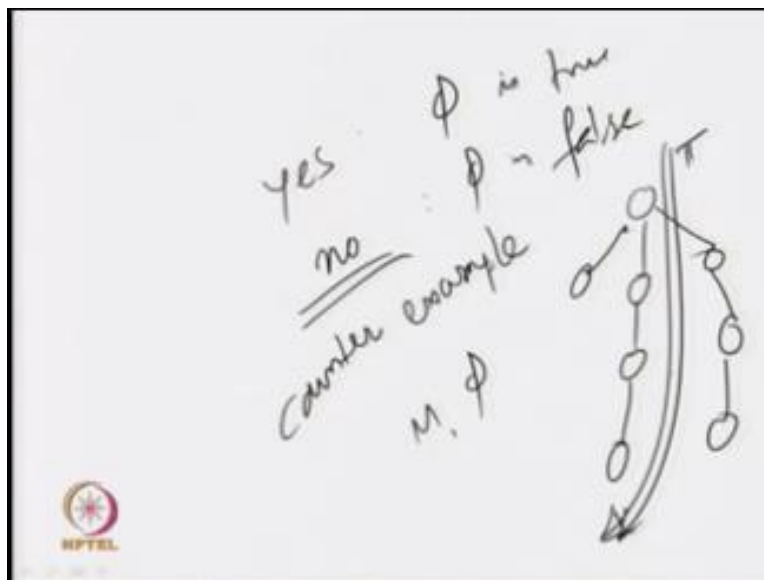
So we are giving the m term model m now the model checking algorithm will be run in the such a way that it is going to give me the state of step I can say that $S_i S_j S_k$ like that some states we are having in which state this particular formula ϕ is true, so we can think the model checking algorithm in this particular levels. So one case first case what we are thinking we are giving a particular state and we are going to check whether ϕ holds or what we are not.

In a second case what we are thinking that I am giving a model and formula it will give me all the steps where this particular formula is true so if we know the method for one the second one can be always direct like that so if I am giving this particular method say I am giving a model m and formula ϕ and it is going to give me all the states where ϕ is true now if I said that I am giving a model m formula ϕ and a particular state say s_0 whether this formula ϕ holds s_0 or not.

So this method is returning me all the particular states where this particular ϕ is true so I am going to check whether this s_0 is a member of this particular return safe or not. If it is member of this particular algorithms then we say that yes ϕ is true be in s_0 , on the other hand if I know this particular method then very well I can find out the particular all the states where this particular formula is true because I am going to run this particular method on each and every step of the model and where it is true it is going to give me those particular step.

So just see that these are we see that complement to each other if I know one the other can be direct and this models checking algorithm is having another advantage also .

(Refer Slide Time: 51:43)

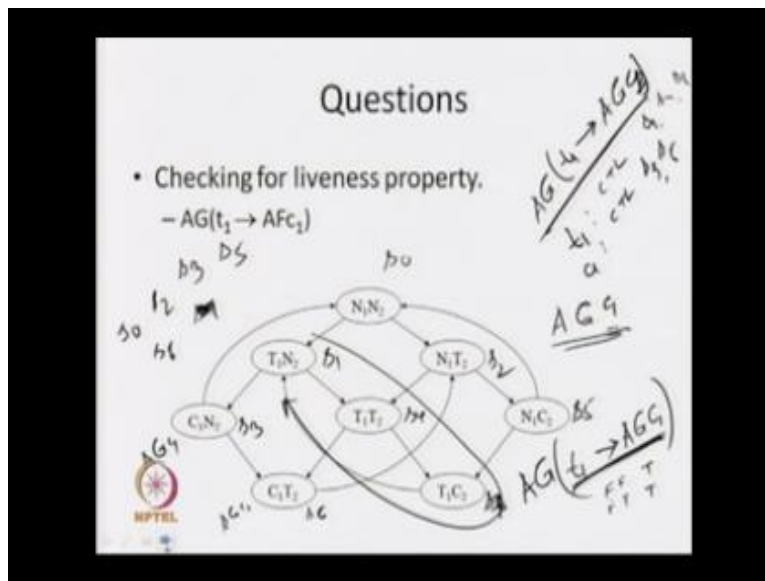


Or another futures also what happen if it I do not say it going to give me either s if ϕ is true and it will going to give me answer no it is ϕ is false okay that formula is false, now when it is no it is false generally that models I got generate the counter example also what is this counter example with counter example it give it says that were this particular formula is false.

Basically I am having giving a model okay so in this particular model that I am giving a formula say ϕ . If ϕ is not true then it is that it is not true in this particular part and then it will give in this particular exhibition place. This particular part says that your particular formula is not true in this particular part Okay.

So that means it is giving a feedback to the designers and the formula is not true that is some problem with my design I have to revisit my design and I have to redesign it but it is the model say it is giving me some clue that my error is some were in this particular execution part. So I can concentrate over here and I can now try to redesign it in such a way the formula because true in my model. But when I am trying to see along this part then some error may occur in so I must be careful about those particular issues also.

(Refer Slide Time: 53:32)



But protocols are giving me some hints so that I can quickly fix my parts okay. So this is the issues of a model in our next class we are going to discuss about this model checking and got him okay. Now look for some questions we are taking about this particular visual expression problem. Now I am going to look for the liveness property checking of this particular liveness property. How I am going to check it or how we are going to check it now you can see that I am

having $AG\ p1$ implies $AG\ q1$ so this the CTL formula for the properties we are given. So here $p1$ is a atomic proportion.

So it is a CTL formula $c1$ is a atomic proportion so it is also CTL formula since they are atomic proportion so we know that evolving function and we know that states where this particular formula or the CTL formula are true. So form leveling itself we will say that this is I can say now in a numbering so this is a 0, $s1, s2, s3, s4, s5, s6, s7$. So $t1$ is true in your $s1, s4,$ and $s6$. $S1, s4,$ and $s6, s1, s4, s6$ so $p1$ is true over here which $c1$ is true you will find that $c1$ is true in your $s3$ and $s6$ okay. So these are the CTL formulas and we know that the true value on this particular case.

Now what is the next formula that we are looking into so this is a atomic proportion so these are the CTL formula now the next formula will come as your Ag all part mobility $c1$. Since $c1$ is a your distance what you call $c1$ is CTL formula so $AG\ 1$ is a atomic. So in all part globally whether it is true or not so if you look into it then you will find that $AG1$ is not true over here so basically if you look into it all part globally. You will find that as per our schematics $c1$ is true over here.

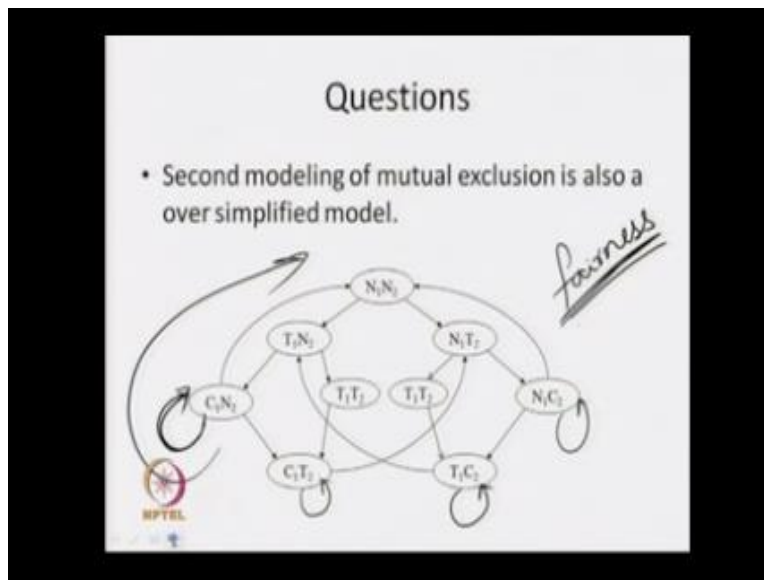
I can say all part globally $c1$ is true over here all part globally $c1$ is true over here okay. So these are the issues we are having in this particular case the next CTL formula will come as your $T1$ in + $AG\ c1$. Now in this particular case we see that if $t1$ is true then the $AG\ c1$ must be true. So if $t1$ is false then obviously this formula is false because false we know that in case of simplification if it is your say false and false it is going to give me true and false and true also give me is true.

So if $p1$ is false then this particular formula is true in this particular case if a loop into the $t1$ implies $Agc1$ then it will be true in $s0$ because $t1$ is false it will be true in your $s2$ it will be true in your $s3$ because your $t1$ is false and it is not your so it will be true in your $s5$ it will be true in your $s6$ and it will be true in your $s7$ okay. So this is your $s7$ because oh! Sorry it is not true in your $s7$ because if $t1$ is false then only this is true. So these are the steps that you are having 1, 2, 3, 4, 5, in this φ step this is true.

But now we have to consult about this particular tree steps what will happen to $p1$ implies $AG\ c1$ you see that $p1$ is true then in all part mobilizing will be true so already we have seen then in this particular part no where I am going to get $c1$ so in this particular tree step these particular formula is false. Now we know that leveling of this particular formula and I can look for the complete formula $AGc1 + AGc1$. Now you itself see in this particular what happens is we are going to look for the true then for these particular formula.

Then must know the rules of each and every formula so in your model checking algorithm, we are going to do in this particular look for each and every false formula. One is then we look for the main formula okay.

(Refer Slide Time: 58:26)



So another one is a that we are having a second model explanation where I have bread this particular formula step $t1$ to $t2$ to two different time step $t1$ and $t2$. And I am saying that the second modeling of visual expression is also simplified model okay. why I am saying this over simplified because you itself see that if someone is entering in the critical section I am saying that after that it is going for non critical section.

Now how many times it is going to be in this particular critical section this particular model is not going to depict your information. For how long it is going to be in this condition time step if it is time step you can go to the c2. Now how long will be in this particular critical step c2, so if I am going to give this particular information basically I can give some sort of this particular self proved formula like that it says that remaining over then it will cover. So once I give this step of self proved then what will happen again it is going to find out my rudeness property because now I will be reminding in this particular case.

So that is why I am saying that the second modeling is over simplified but if I come up with this particular self proved I said it will remain in the form for more number of times and it will come out then what will happen is the we try to look for checking those particular property thus it will be false. But what we can do we can use some fairness of model checking to result the issue. Either I can come up with a simplified model or I can try to depict the information.

Where it may know pilot sound property what will happen we are going to use some fairness execution in this particular model or we are going to use some fairness constant. What is this fairness execution basically now as diagnose will be equality it remains in this particular step forever. But we know that in any situation it will enter in the critical section it will do job first required which will take some amount of time eventually it will come out that means we are going to look for a fair execution for that only it will not remain over here infinity.

(Refer Slide Time: 1:01:05)



Eventually it will come out after completion of job it will come out that means what we are going to say that we are going to look for a fair execution that means I can think some model okay, okay. So now you itself see that in this particular loop the model will remain finite okay that means if my system will remain in this particular loop then I going to have some bad behavior but in fairness what I am going to set up if my system is correct I know then what will happen eventually it is going to come and from this particular loop.

And eventually it will flow this particular part that means we are going to look for fair execution we can somehow omit this particular infant loop so try to omit this particular part and we will see that eventually it will come out this particular loop and it is going to set. So in this particular case in this particular fairness constant we can check for this particular property because while I am designing a system it is coming actually I cannot break this particular loop.

Because it may be here but my models are giving me some problem or it is having some error so I will say that eventually my system will come from the loop to check my property in fair parts only fairness constant into look for the fairness that means my system will go to the fair parts

only so like that we can look for the fairness and go for a model in the same system only without braking this particular problem okay.

With this I will stop my lecture today so in next class we are going to look for the model checking algorithm.

Centre for Educational Technology

IIT Guwahati

Production

Head CET

Prof. Sunil Khijwania

CET Production Team

Bikash Jyoti Nath

CS Bhaskar Bora

Dibyajyothi Lahkar

Kallal Barua

Kaushik Kr.Sarma

Queen Barman

Rekha Hazarika

CET Administrative Team

Susanta Sarma

Swapan Debnath