Embeded Systems - Design Verification and Test

Dr. Santosh Biswas
Prof. Jatindra Kumar Deka
Dr. Arnab Sarkar
Department of Computer Science and Engineering
Indian Institute of Technology, Guwahati

Lecture - 19
Syntax and Semantics of CTL

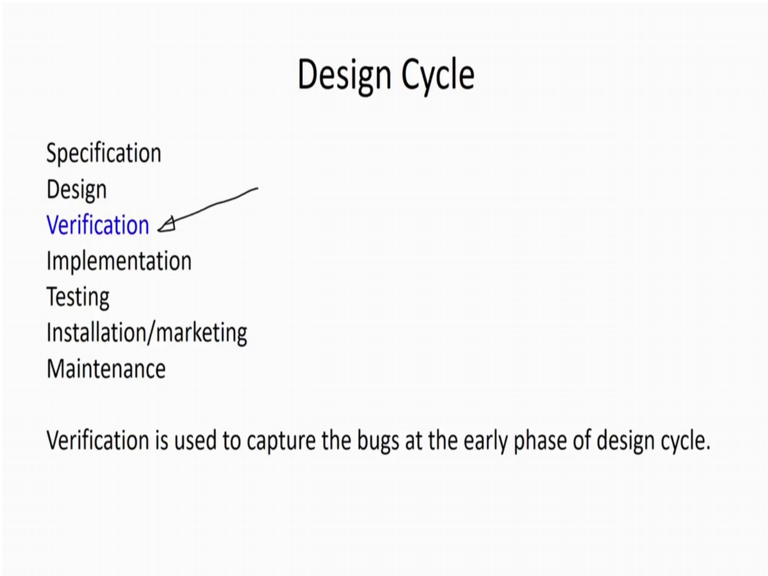
Welcome back to the online course on Embedded System – Design, Verification and Test. By the name itself we can say that there 3 component in this particular course: one is your design, second one is verification and third one is test. I think by the time Dr. Arnab Sarkar might have covered the design issues of embedded system. I am Jatidra Kumar Deka, now we are going to discuss about the verification part and finally, Dr. Santosh Biswas will discuss about the testing part of embedded system. So, now onwards we are going to talk about the verification of embedded system.

So, in general we are going to talk about the verification tools that is needed to verify some system design, ok. Now, the first component for your verification is one logic by which you can specify the property of the system. So, in this particular course we are going to talk about the temporal logic. So, temporal logic will be used as a property specification language and we are going to describe the properties of the system and that property will be verified on the model that we are going to design, and model is nothing but the abstraction of the system that we are going to finally implement.

So, if you look in the design cycle, the basic cycles or basic steps for your embedded system basically falls on those particular categories; one is a specification depending on our requirement depending on the user state we have to specify the system behaviour. So, system behaviour has to be specified in such a manner that it is going to satisfy all the requirements specification. Once the specification is ready then from the specification we are going to have that design. So, one design is completed then we will go for the implementation of the system. So, in case of embedded system we having two components hardware well as software.

So, we are going to implement all the components, we will integrate them and finally, we are going to get that system. One system is ready then we will go for a testing, that just to give it a sense that whatever we have implemented it is going to work correctly and finally, it is installation and marketing and later on the maintenance phase and operational come into picture. So, these are the basic design circle.

(Refer Slide Time: 02:59)



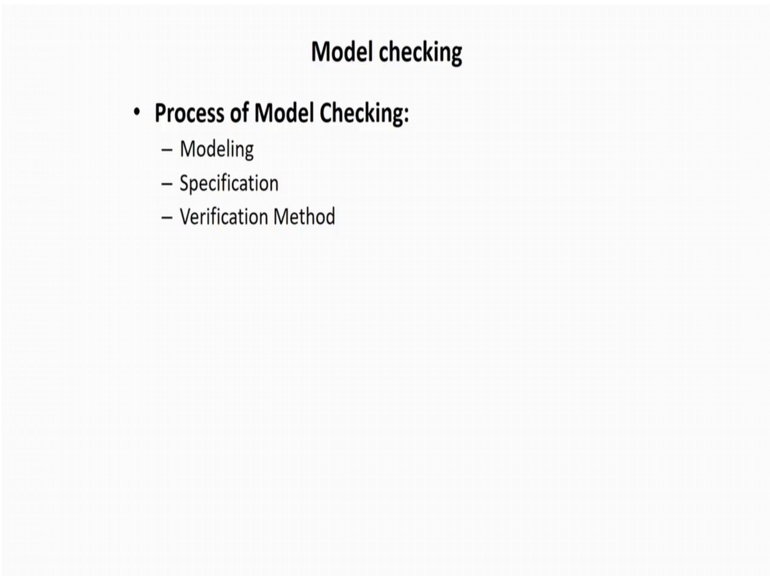
Now, what is the challenges in this particular embedded system design? Generally, what will happen? When we going to design a particular system it may happen that design may not be perfect due to some of the regions, and we lead go into lead some errors, those errors are going to be treated as a bug of the system and as a system designer team we have to remove those particular bar for the proper functioning of the system. So, it says that bugs reported at early phase of design incur less cost for debugging. Now, what will happen? Say in this particular design cycle of try implementation we are going for the testing.

So, if during testing if some errors reported then we have to fix those particular error that means, again we have to start from the beginning, we have to look into the specification whereas, (Refer Time: 04:03) correct or not, then we have to look for again redesigning it, then again implement, then testing. So, one order basic emphases is that after design if possible we are going to capture the error that remains in the design phase and once you can capture the error, then what will happen? Again we can go back to the design phase and we can redesign it before going to the implementation.

So, we are going to capture the error at the early phase of design cycle. So, for that in the design cycle we have incorporated one more step which is known as your verification, ok. So, in that particular verification we are going to check whatever design we are going to come up that design is going to fulfil the requirement. So, design the abstract level we are going to do it. So, for that verification part what are the components required, what are the things needed we are going to discuss in this particular series of lectures.

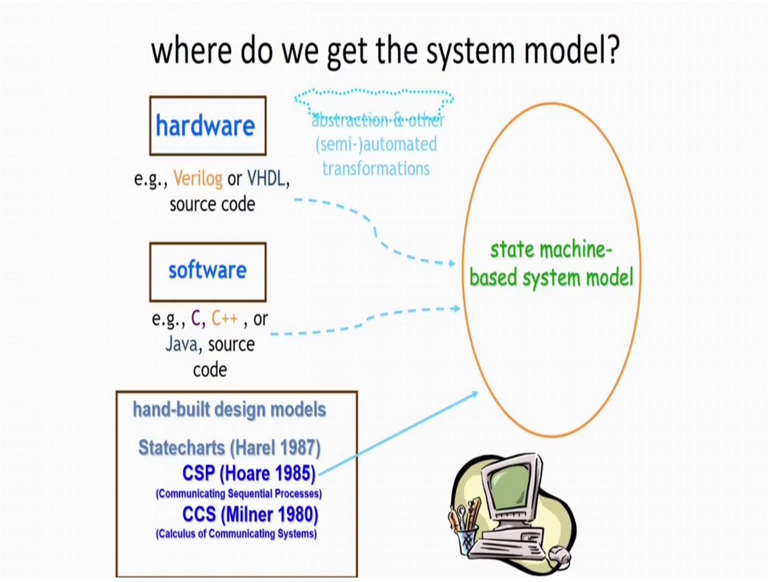
So, there are several tools and techniques to go for the formal verification or a system design. So, in this particular course we are going to look for a particular method which is known as your model checking. So, model checking is a property for a verification method. In this particular method what generally we used to do? We are going to check the properties that are going to be satisfied by the system that we have designed, so that system design or you can say that system or a model of the system that we are going to design. So, after the design phase we are going to look for the property verification or going to check whether all the specification are going to made by our design or not.

(Refer Slide Time: 05:53)



So, in model checking basically we need 3 different component, one is called modeling. So, this is going to represent the entire system with the help of a model we having different formalism. For model checking we need one particular formalism we are going to discuss with about that particular formalism. Then once the system is ready then next step is we have to identify what are the properties that need to be satisfied by the our system we can say these are the specification of the system. So, we are going to have the model our system then we are going to have all the specification that is required to be satisfied by our system. Once we have these two component then we need an verification method we are going to apply this particular verification method to check whether specifications are correct in this particular model or not. So, basically these are our steps or requirement involved in the model checking technique.

(Refer Slide Time: 06:57)



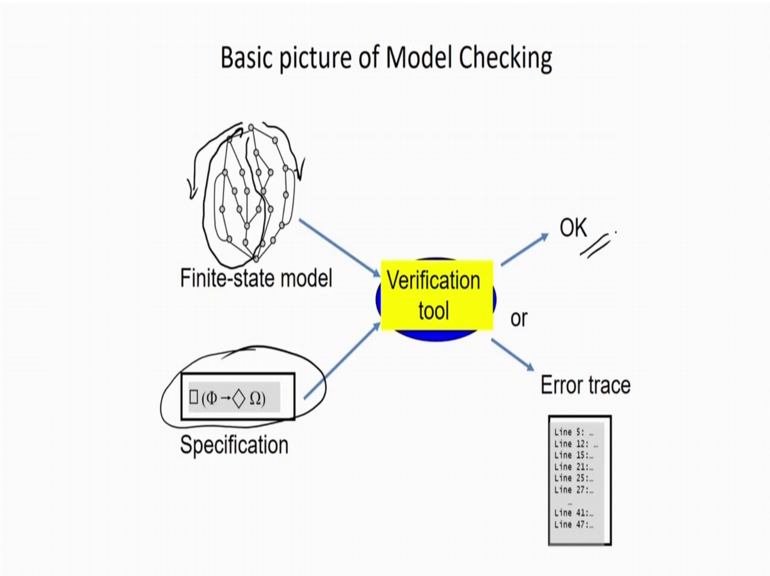
So, in pictorially we can see our system in this particular first we need to have our model. So, after having the specification and doing the modelling then what will happen? We are going to have a system model. Now, how we are going to step a system? In embedded system we are having both the components software as well as hardware. So, all the hydro components will be described with the help of some hardware description language you may be knowing some of them as your Verilog or VHDL. So, implementation of the system will be described by those particular hydra description because if the components are hydroid.

We are having some software component for that software component generally we may use some high level language may be C, C plus plus or Java or as per our recommend you can choose any programming language. So, hardware component and software component will be implemented with the help of respective programming languages. Then after that we are going to abstract the model form those particular program. So, abstraction in some cases it may be automatic or sometimes you have to do manually or sometimes we have some semiautomatic devices or methodologies to get the model of the system. And generally the model of the system is represented with the help of a finite state machine, ok, it is a finite state model because what about may be the system how complex it is generally we are going to have a finite number of states and we are having transition between those particular states.

So, these behaviour that finite state model is going to give us that complete behavioural model of the system. So, once you get a model then we are going to look for a specification. Apart from that we are having some other formalism also to build a model, one is known as your state sort which is proposed by Harel in 1987. So, it is also extensively used to describe our model. So, (Refer Time: 09:10) after getting a complex vision of the system instead of going for your software or hardware design we can describe the model or behaviour of the system with the help of state chart.

Another formalism is there which is called CSP, communicating sequential process. It was proposed by Hoare in 1985. A manometer we have CCS calculus of communicating system it was proposed by Milner in 1980. So, if we need we may use those particular formalism also to get the model of the system. When the model is ready then we have to see what are the properties that need to be satisfied that particular model.

(Refer Slide Time: 09:56)



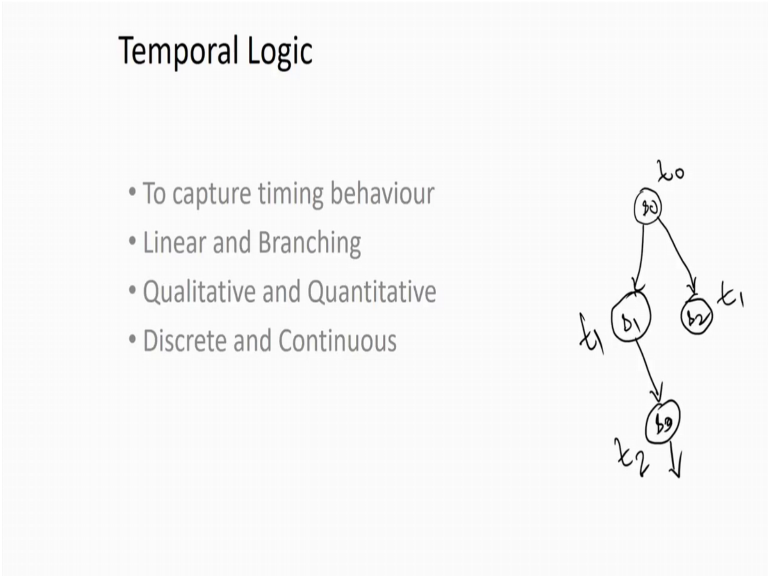
So, in that particular case we are going to have the finite state model of the systems are basically these are a difference state a system can have, and these are the possible transition we are saying, and this is a complete state transition behaviour of the systems.

So, we are going to have this particular model and the specification have to be given in some specification language which is having a proper syntax and those syntax has been given a proper semantics. So, here we are just giving some symbols over here that box phi implies diamond psi. So, this is a one kind of language to which you can specify the properties.

So, this finite state model and specification will be given to the verification tool and in that particular course we are going to look model checking is a verification tool. Once you give this particular specification and finite state model to verification tool than verification tool will give us a result as ok, keep the property holds in this particular model, ok. So, this is one way you can look into it. So, if property is satisfied then (Refer Time: 11:14) will give us, ok. If the property is not satisfied in a model then what will happen? It will give us an error trace it says that if we go through this particular error trace, then if you follow this particular trace then your property is not verified. It may happen that, so we are having several transition apart from this particular state.

So, it gives something like that if the property is not satisfied say in this particular transition, then it will give us this particular error trace that means, if we are moving from state in this particular part then the given property is not satisfied. So, in that particular case as a design them what we can do? We can concentrate on those particular part to fix the bug once you fix the bug then what will happen I can be repeat this particular process of verification finally, if it says, that means, the given specification is correct.

(Refer Slide Time: 12:17)

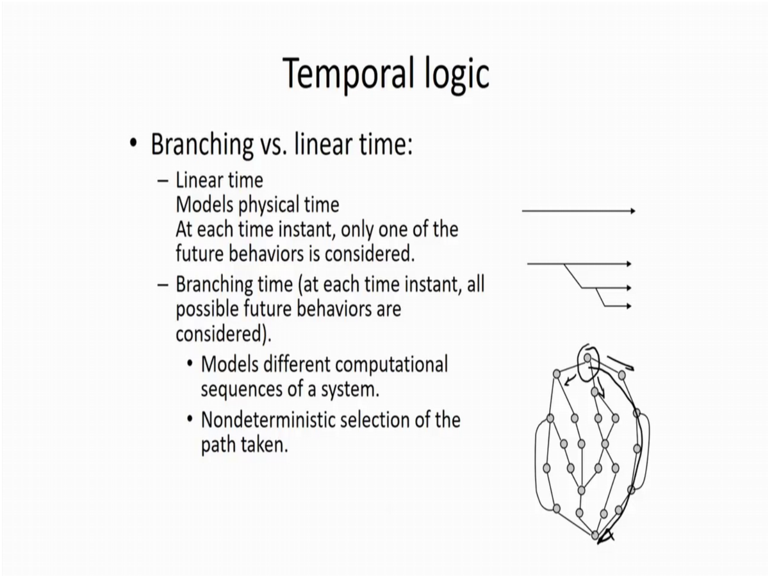


So, in this particular tool or method model checking general we use temporal logic to specify the properties of the system. So, in the previous slide whatever we have written this type manager specification this is a kind of temporal logics formula. We are going to discuss or going to see: what are the meaning of those particular symbols.

So, in case of temporal logic what will happen it is going to capture the timing behaviour of the system, because if it is a state transition systems I am in a state s 1, s 0 and I am having a transition to s 1 or maybe it can have transition to s 2, from s 1 we can go to s 3 then what will happened this is the transition behaviour of the system. So, in that particular case what will happen? If in time t 0 the system is in your s 0, then in time t 1 it can go to state s 1 or it can go to state s 2, and once it comes to us state s 1 then in the next clock cycle system can go to s 2. So, this is that timing behaviour our systems. So, temporal logic can capture this particular timing behaviour.

And when we are going to look into the timing nature of the time it can be look into two different aspect, one is your linear and second one is branching. Also time can be treated as your qualitative and quantitative, and we may have that discrete notion of time or continuous notion of time.

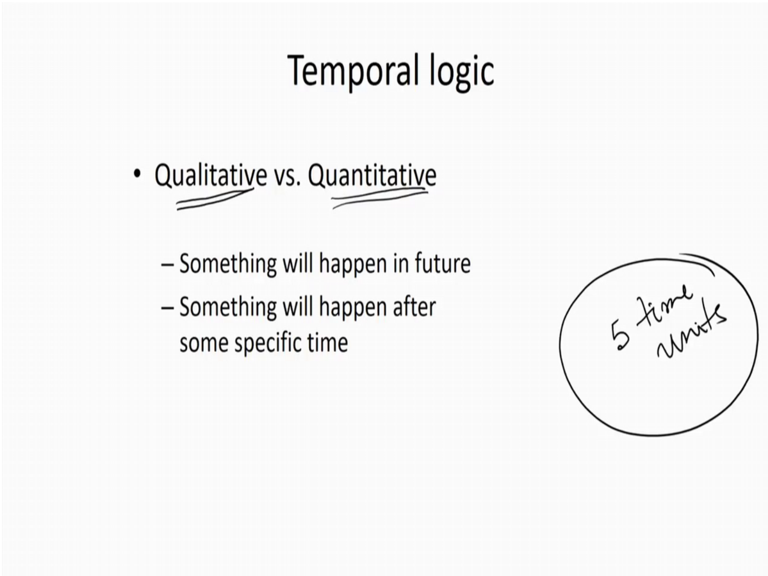
(Refer Slide Time: 14:03)



So, in case of branching and linear what will happen? In case of linear we said that time progresses in one particular transition or one particular execution trace. But in case of branching time logic what we consider, that from a particular step we are having several possibilities system can transit to several possible behaviour and, but at any particular instance of time it will take only one next step, ok. Thus in case of this particular diagram you just see if we are in this particular step then we are having 3 possible behavior of the system. So, time progresses in (Refer Time: 14:50) different direction, but at any particular instance of time it is going to take only one next step behaviour. It cannot go to all these 3 next that behavior. So, depending on the environment or the system variable either it is going to take this particular part or it may take the (Refer Time: 15:10) part.

So, in this particular case the timing or behavioural of timing is branching in nature. So, we are going to see all possible cases and we are going to reason about the system in possible direction. But in case of linear time at any a particular instance of time we are going to look for one direction only one execution trace in one time direction only. So, at that time we are may ignore the other two behaviours, but later on once a he look the behaviour in one direction then we can go for the second direction. In linear time logic we just look for one particular execution trace, but in case of branching time logic we are going to look the entire system as a whole and we see the behaviour in all possible direction. So, temporal logic is having to define one is your linear time, second one is your branching time.

(Refer Slide Time: 16:08)



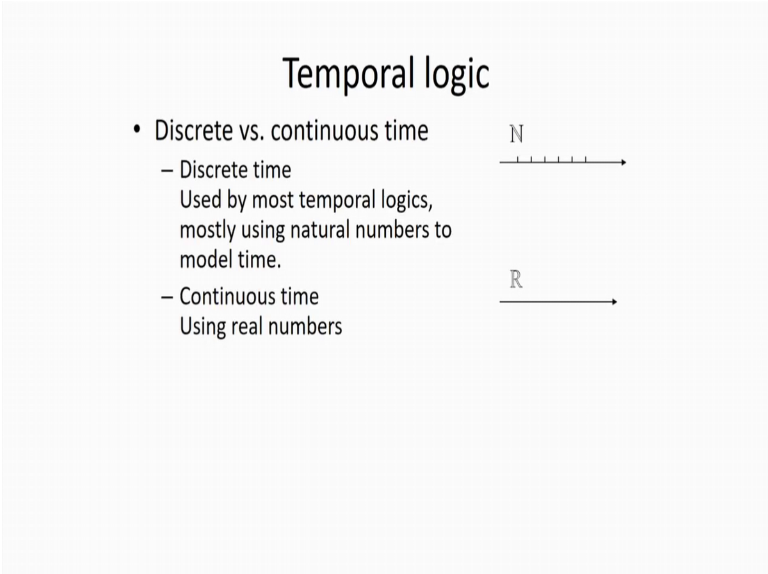
It is qualitative versus quantitative. So, it says that something will happen in future, something will happen after some specific time. So, you can say the simple example generally is to say in our network protocol. If we send the message then eventually the sender should get the acknowledgment from the receiver. So, this is a property that needs to be satisfied by the protocol in our network system. So, you say that if we send a message then sender should eventually get that acknowledgment message then only and I will be knowing that message has been delivered properly.

So, if we are going to special a property in that particular way then we are going to say this is your qualitative in nature. We said that eventually or in future it should get the acknowledgement. But in quantitative nature what level are you going to specify the quantum of time. So, in that particular case what we should say that if sender sends a message then sender should get the acknowledgment after all 5 unit of time. So, here we are not saying eventually it should get the acknowledgment, but we are saying that after all 5 time units.

So, what we are saying? We are giving some quantitative measures so that means, sender will wait for 5 time units and within 5 time units if sender does not received the acknowledgement, then what will happen? In most of the protocol we (Refer Time:

17:44) the packet again. So, these are the two ways you can specify the timing behaviour, one is your qualitative nature, second one is your quantitative in nature. Now, of course, we can discuss, we will see what the behaviours actually one after another.

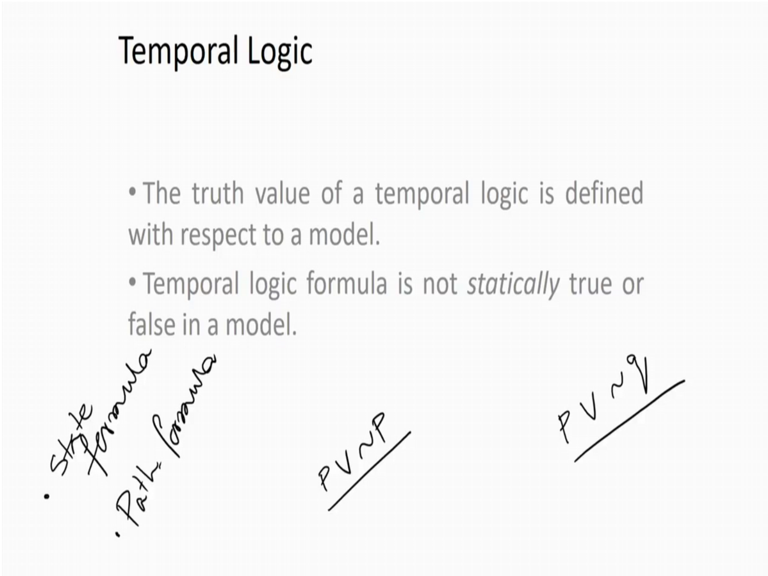
(Refer Slide Time: 18:00)



When we come for a quantitative one that means, we are saying after 5 unit of times or say after 10 units of time. So, this quantum of time can be can be your discrete in nature or it can be continuous in nature. So, in case of discrete time domain what will happen? We are going to give discrete time minutes of tau one time unit after 2 time unit, after 5 time unit or maybe after 10 time unit, but in continuous time generally we are going to new product continuous time line. So, it will be specified in real number. So, we can settle after 5.5 time units or say 5.5 second like that.

But if we walk in the digital domain in most of the cases you will find that the time that we capture in our digital system is always controlled by a master clock and the nature of timing discrete in nature. So, our emphasis will be on the discrete time system. We may not give emphasis on the continuous time domain because in continuous time domain system become very complex and verification of properties even more complex. But when work in the discrete domain it is relatively easy, so time we ever can again look into as a discrete in nature or continuous time.

(Refer Slide Time: 19:30)



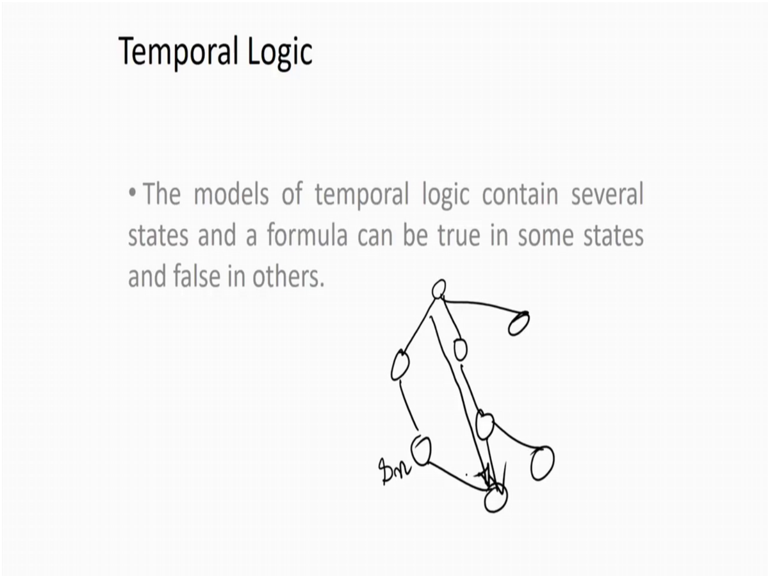
Now, in temporal logic we are having a provision to capture the timing information, this is the addition part in the temporal logic. It is like your, I think in your classical logic you know about propositional logic, predicate logic or higher order predicate logic. In this particular logic we cannot capture the timing behaviour, but temporal logic we can capture the timing behaviour.

So, in case of temporal logic: what is the truth values of a temporal logic formula? That truth values of a temporal logic formula although as defined with respect to a model. So, if we are writing a temporal logic formula simply you cannot say that it is true or it is false. The truth values of a temporal logic formula is always specific to a particle model. So, in simple example I can give in propositional logic what I can write I can say that p or not top q. So, this is a formula in our professional logics. So, in truth values of this professional logic will be either true or false depending on the truth values of p and q.

Secondly, if I say that p or not of p again it depends on the truth values of p and we know that this is a topology this is always true. But in case of temporal logic formula we cannot say that the formula is true or false, but always you have to keep the meaning truth values of the temporal formula with respect to a model.

Secondly, temporal logic formula is not statically true or false in a model. So, we cannot say that this even temporal formula is true in this particular model a or it is false in a model b, but it depends on the state behaviour of the system or may be transition behavior of the system. So, you having to type of temporal logic formula one we said state formula and second one is your part formula, ok. So, if path formula if we are going to talk about the path formula it says that the temporal logic formula is true in a particular part of the given model and if we say that state formula that means, that temporal logic formula is true in a particular step of the given transition system.

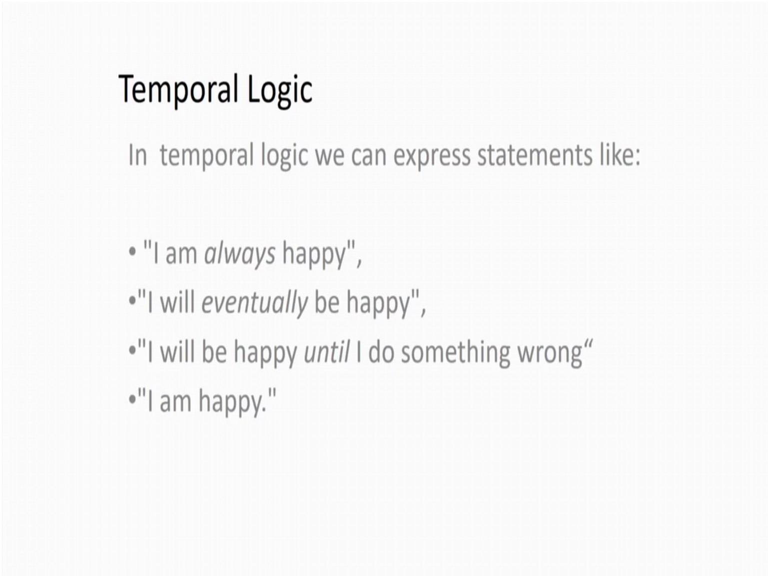
(Refer Slide Time: 22:08)



So, the model of a temporal logic contains several states and a formula can be true in some states and false in some other. So, truth values of the temporal logic formula is specific to a state if it is a state formula. So, for a given temporal formula in a given model the formula may be true in some of the states and it will be false in some other states. Similarly if it is a path formula the formula will be true in some other parts, but it may false in some other part, ok.

So, this is basically now, I can I can give a emphasis on that say I am going to consider a particular model, state transition model something like that. So, if I am giving a temporal logic formula and say this is the step s n I can say that given temporal formula is where a true in s n or false in s n. I can just talk about the state if it is a state formula we cannot say that given temporal formula is true in this given model it is not. Similarly, if it is a path formula we are going to say that whether the formula is true in a particular part or not, ok. So, notion of the truth values of a temporal logic formula is define on a state if it is a state formula or it is defined on a part if it is a path formula.

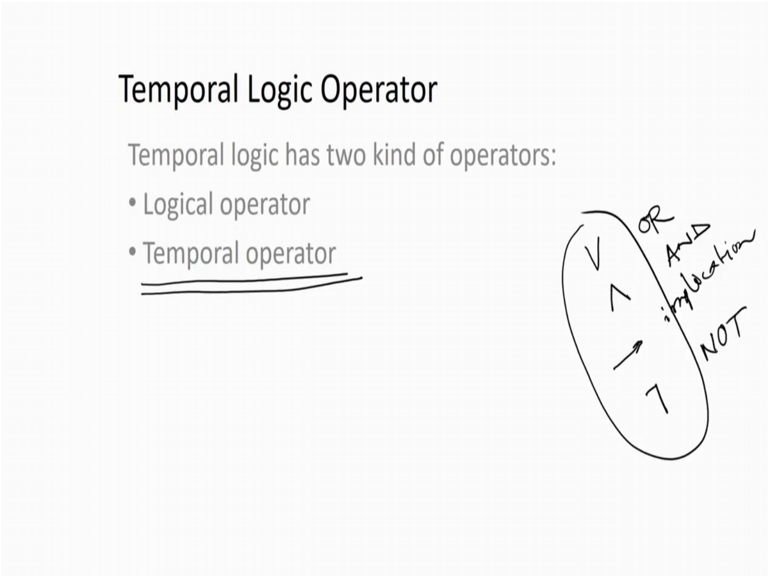
(Refer Slide Time: 23: 45)



Now, what we can specify in our temporal logic? Very simple statement I am giving say I am always happy. So, it talks about I am always happy. So, what I can specify these things temporal logic, yes we I can construct to specify these things or we can say that I will eventually be happy. Now, I am I may not be happy today, but in future or eventually I will be happy.

Another one I will say that I will be happy until I do something wrong. Today I am happy and I will remain happy till I do something wrong because if I do something wrong then I will go to depression or I feel unhappy or I am happy this is the current status, ok. So, these are the things where they are capturing the timing behavior although our lifeline. So, such time of timing can be captured with the help of temporal logic.

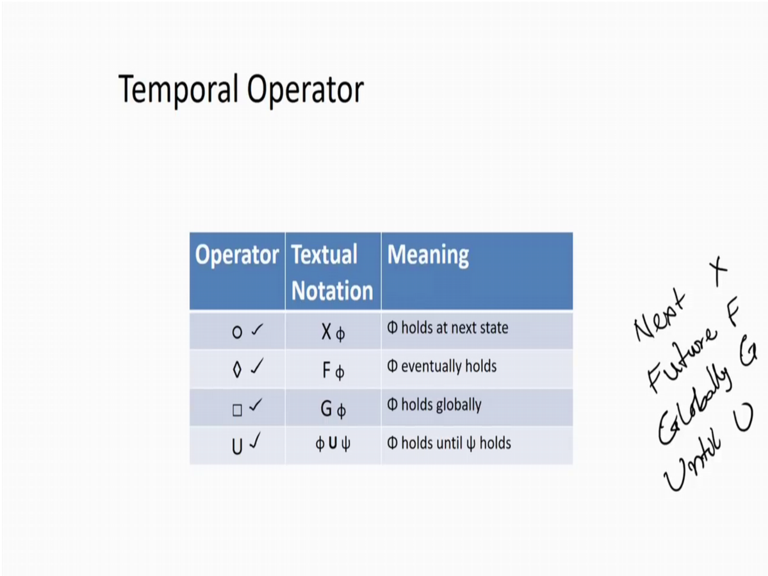
(Refer Slide Time: 24:34)



So, what we have in the temporal logic? Already I have mentioned that we are having proportional logic, we are having some proportional logic operator like disjunction or operation, conjunction or an operation than if then implication, ok. These are the some basic operator that we have you know propositional logic or we can have negation also not. So, whatever logic operator we have in our logic family all those operators can be used in our temporal logic that means, two formulas can be conjuncted with the help of an operation or we can use the disjunction also on two formulas to get at another predict formula.

So, and I think you know the meaning of those particular operators. So, we are going to use the same meaning in temporal logic also. So, we are going to use the logic operators in our temporal logic also. Along with that we are having some temporal operators these are the addition operator that we are having in our temporal logic. So, all the logic operator as well as we are going to use the temporal operator also in temporal logic. Now, we are going to discuss about those particular temporal operators, what are the temporal operators that we have in our temporal logic.

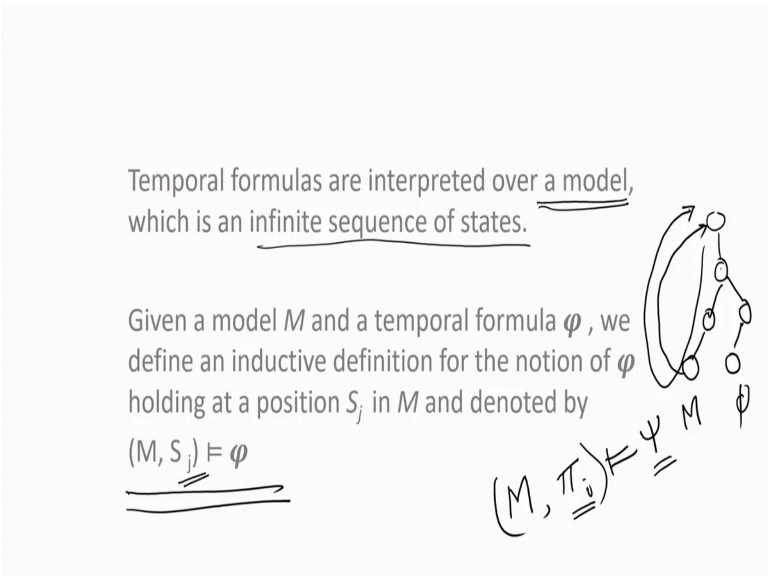
(Refer Slide Time: 26:09)



So, there are 4 basic temporal operators that we have, but we can have we can look for some other operator also, but that can be again implemented with the help of those particular basic operators. So, what are the basic operator that we have in our temporal logic? Here the if you look the historical development of our temporal operators it has given some symbols to represent this particular operator one is your circle, diamond, box and U. So, these are the operators symbols that generally used in a temporal logic, but nowadays we are going to represent those particular operators with textual notation. And in case of textual notation what are the alphabet we are using? It is X, F, G and U; X is nothing but it talks about the next step this is X, F is basically future, then globally is G and one is your until U. So, these are the 4 letters you use.

So, basically if I said that X phi is a temporal logic formula, X phi is also a temporal formula. It says that phi holds at next step. If I say that F phi it says that phi eventually holds or phi eventually becomes true. G phi it says that phi holds globally and phi until psi it says that phi holds until psi holds that means, phi remains true until psi becomes true, ok. So, these are the 4 temporal operators we are going to use in while going to specify or when going to capture the timing notion of our system. So, 4 operators specifically we are having one is your next, second one future, then globally and until.

(Refer Slide Time: 28:28)

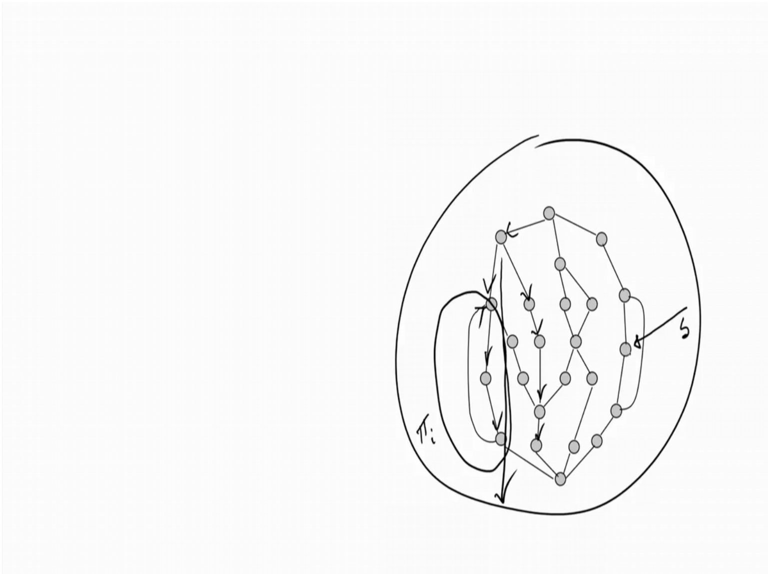


So, now how we are going to get that truth values or how we are going to define it? It is going to define in a model. So, temporal formulas are interpreted over a model, which is an infinite sequence of states, it is having infinite sequence of state, but we are saying be clear about it. We are talking about infinite sequence of states, but we are not talking about infinite states because whatever system we are going to model in real they will (Refer Time: 29:02) that all are having finite steps, but that finite step some of the state may repeat forever, ok. So, we are going together infinite sequence of state. Simple example I can give say, in a model I can have such type of behaviour. So, you just see that if I going to have this particular execution phase I am having 4 steps only, but system can go to this particular part over and over again. So, we are going to get an infinite sequence of state.

So, temporal formulas will be interpreted in the model where we are going to get some infinite sequence of state. So, how we are going to define it? Given a model M so, this is the model M and a temporal formula phi, we define an inductive definition for the notion of phi only at a position S j in M and it is denoted by M S j model phi. That means, in the system M in the state S j phi (Refer Time: 30:11) or I can say that M S j model phi. So, if we are using this particular S j that means, we are going to talk about the state formula.

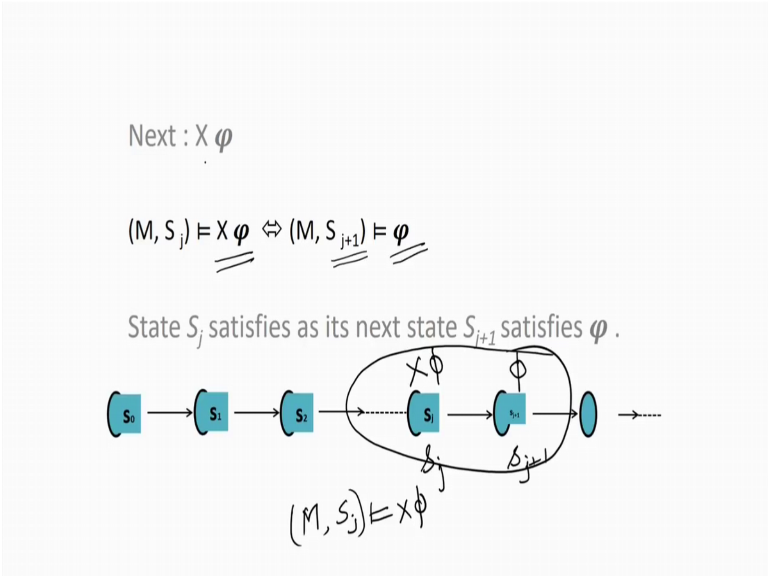
Similarly we can say that in the model M in the path pi i each models say psi in that particular case psi is a path formula, and truth values of the path formula is defined on a particular path pi i. So, this is the way we can always interpret that we are going to look for the truth values of a temporal formula in a model and in the model we are going to look for a infinite sequence of states. So, again two issues are there, one is a state formula, second one is path formula. Truth values of that path state formula will be defined over a path.

(Refer Slide Time: 31:07)



So, this is the some state transition diagram, ok. So, state can have this particular behaviour I can say something like that. So, you can talk for state formula, you can talk about a particular state say S j, or I can talk about the truth values of a formula in a particular path say pi i, ok. So, this is specifically path formula which will be defined over this particular path and this is state formula which will be different in a particular state truth values will be defined on this particular state.

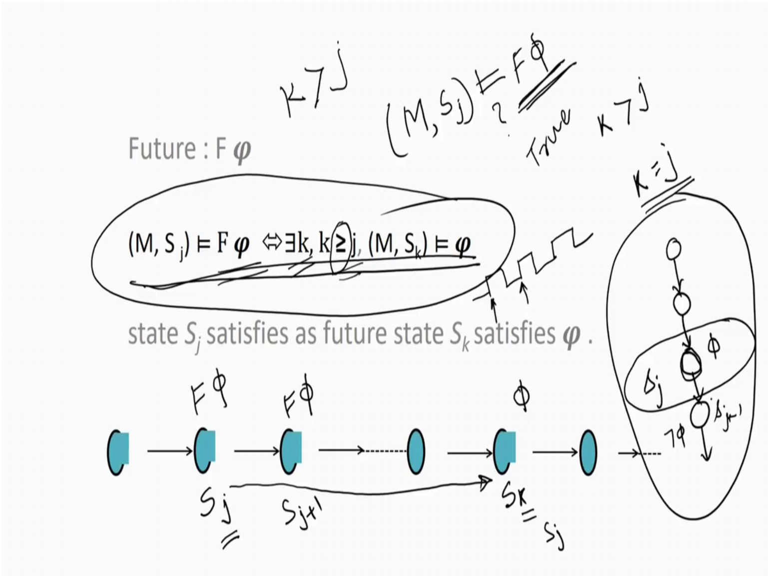
(Refer Slide Time: 31:56)



Now, what is the meaning of this particular operator when I say that next operator or X operator? So, it says that X phi what does it means? That means, it says that if I am in a particular state whether something holds in the next state or not, ok, so that truth values is defined like that. So, M S j it models X phi, where X phi is a temporal formula next phi when we say that M S j whole X phi if or provided if M S j plus 1 models phi. So, this is a next step.

So, in this particular case you say this is my state S i, S j, this is the state S j plus 1, ok. So, if we say that phi holds in this particular state S j plus 1 then according to this particular definition of next operator we can say that X phi holds in the state S j. So that is why I am saying that model M S j model X phi or X phi is true in the state as if we are having such type of behavior in our system that means, in the next state j plus 1 phi is true then in state S j we are going to say that X phi is true. This is the way we are going to define that truth values or by temporal formula which involved this particular next operator. So, this is simple meaning of next operator.

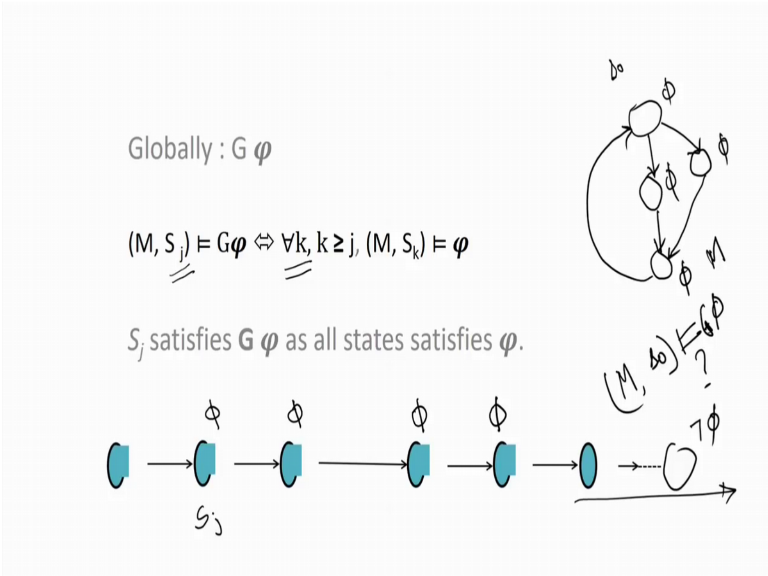
(Refer Slide Time: 33:50)



Similarly, the next operator is future or F phi. So, when we say that this particular formula F phi will be true in a particular state say S j. So, it says that in a model M in S j holds F phi that means, M S j model F phi this is future of phi this is a temporal operator provided or even and if we are going to get some k here exist some k, where k is greater than equal to j, and M S k models phi, ok. So, we are going to have some k S k where phi holds then we say that in a given state j S j F phi will prove provided k is greater than equal to j. So, in this particular case I can say that this is my state S j and say this is my state say A k and instead at in A k if phi is true (Refer Time: 35:02) labelling it phi it indicate that in this particular step phi is true, ok.

So, in that particular case if I look into that S j we will find that if I go to this particular execution place somewhere in future we are getting a state where phi is true. So, in that particular case I said that here in this particular step F phi is true. So, similarly if I look into this S j plus 1 again you will find that F phi is true in j plus S j plus 1 also because from this particular step if you go through this particular execution (Refer Time: 35:42) in future I am going to have the state where phi is true. So, in this particular case also you can say that whatever state we are having before this A k in all the state F phi is true. So, this is the notion of the future.

(Refer Slide Time: 35:57)

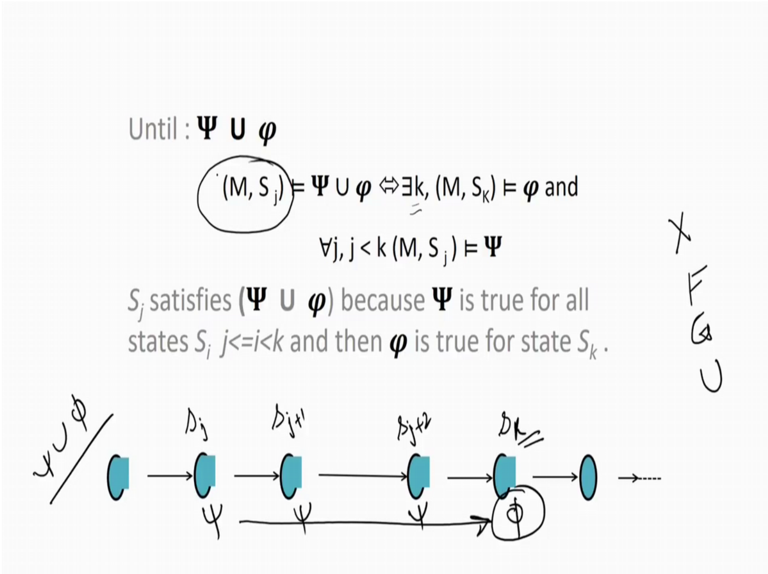


G phi globally it says that whether globally phi is true or not. So, how we are going to define it if we are going to look for one particular state say S j in a model M when we say that G phi holds in your model M in state S j if a provided for all k, where k is greater than equal to j we are going to say that M S k models phi. So, in all state wherever we go from this particular state if everywhere k is phi is true we said it G phi is true at the particular point that means, you can say that phi globally true.

Now, in this particular case if this is my S j if I say that phi is true over here, phi is true over here, phi is true over here, not. Now, can you say whether G phi is true at the particular point it will be slightly difficult because phi is true up to this particular point, but we do not know what is going to happen in future states whether phi will remain true or not. So, in that particular case I cannot say that G phi holds in this particular state, but if I am going to have another system. So, in this particular case what will happen? Now, if this is the state say s 0 and say this is model M whether in the model M in state s 0 whether it phi holds or not. So, in that particular case what I can say that s phi holds in G phi holds or not. So, I can say that in state s 0 G phi holds because wherever you go which (Refer Time: 38:20) you take in all the states in future phi is going to be true.

So, in this particular model I can say that yes M s 0 model G phi, but here it is slightly difficult to say because I can have one state somewhere here because I am talking about the continuity answer phi does not hold over here. So, in this particular state I cannot say that in as G phi holds. So, in all possible state if phi is true then we said it globally phi is true and this is true for all the states.

(Refer Slide Time: 38:53)

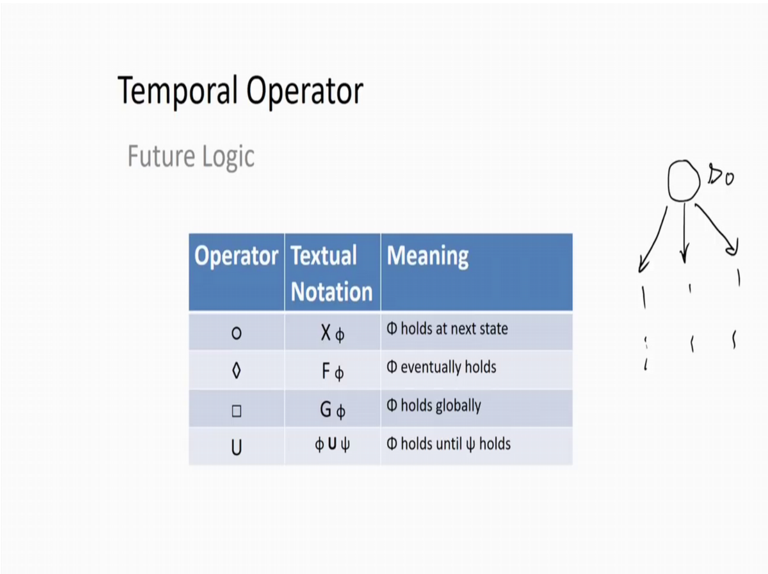


Last operator is your until, psi until phi. So, what is a notion it says that in the state S j of M it satisfies psi until phi provided psi holds in a state and all the state previous to that particular state if phi holds psi holds then we say that psi holds psi until phi holds. So, here you just see it says that in this particular state psi until phi holds in S j provided there exist some k, where M S k models phi and in all j, where j is less than equal to k M S j model psi, ok. So, in that particular case we said that in the state S j psi until phi is true.

Now, come to this particular point say if I am going to say this is say phi holds over here and say this is the state S k, and say this is your S j, S j plus 1, S j plus 2 like that. So, if all the state say psi holds then what we are going to set up? From this particular state S j in this execution trace we are going to get one state where phi holds in state as k and all the previous states whatever we have encountered in all those particular step psi is true and in that particular case we are going to say that in this particular state S j psi until phi holds or psi until phi true. So, something remains true until some property becomes true. So, this is the notion of until operator.

So, in that particular case if this is the scenario then we are going to say that in the particular step psi until phi is true. So, these are the meaning of those particular temporal operators. What are the temporal operators we have discussed next state, future, globally and until, and these are defined in a model. So, this operator can come as a state formula or it can appear in a path formula. Here when we are going to when we are going to look into truth values if it is a state formula we are going to talk about the truth values in a particular state, if it is a path formula then we are going to talk about these truth values of the formula in a particular path. But here the way we are defining basically we are defining the truth values of the temporal operator in a state, ok. So, these are the 4 basic operator that we have. Now, say whatever we have discussed or that operator basically these are future in nature.

(Refer Slide Time: 42:34)



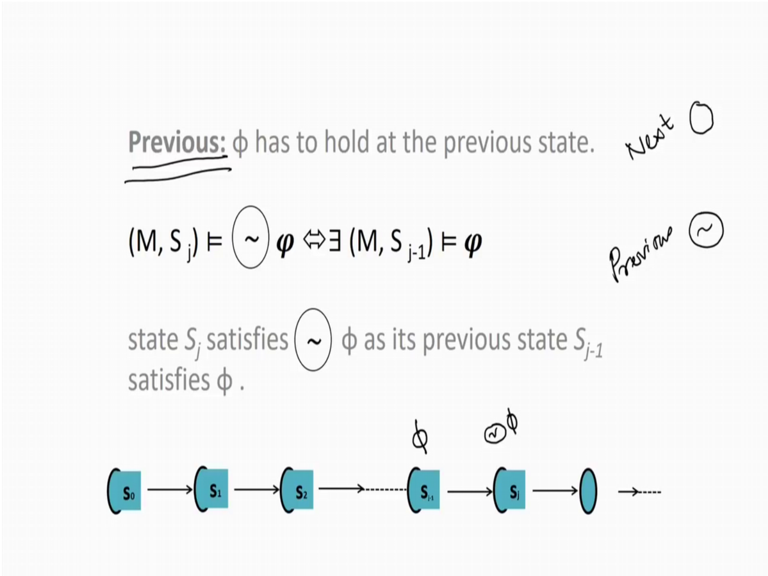
So, we said this is future logic because we are in a current state and from here we are going to reason about the system in the future that means, how the system is going to be a in future. So, this is basically future logic. So, similarly we are having a notion of past logic also.

(Refer Slide Time: 42:54)

So, in past logical also we are going to have the corresponding operator with respect to the operator that we have in your future logic. So, in future logic we are having the next. So, in past logic we are going to get an operator previous. In future logic we having the operator of future so here we are going to have eventually in past, in future logic we are having global operator so in past logic we are going to have globally in past and we have the until operator so in past logic we are going to have back to operator.

So, if we are going to region about a system from a current state and how the system is going to behave in future then we are going to use the future temporal logic. But we are coming to a particular state, but if we want to know what was the behavior of the system in the past then we have to use the past temporal logic basically we have to use the operator that is available in the past temporal logic.

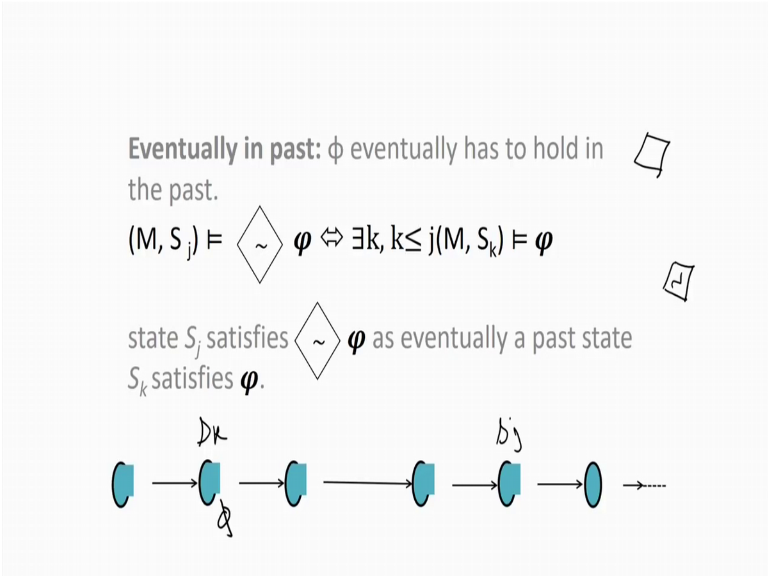
(Refer Slide Time: 44:07)



So, this is basically similar to your future temporal logics. So, in case of future temporal logic we are having a next state, but in case of past temporal logic we are having the previous, ok. So, in the temporal logic say we are saying that (Refer Time: 44:27) is given as a next step. So, if we write whole and inside that if we write the tilde than it is going to say this is the previous.

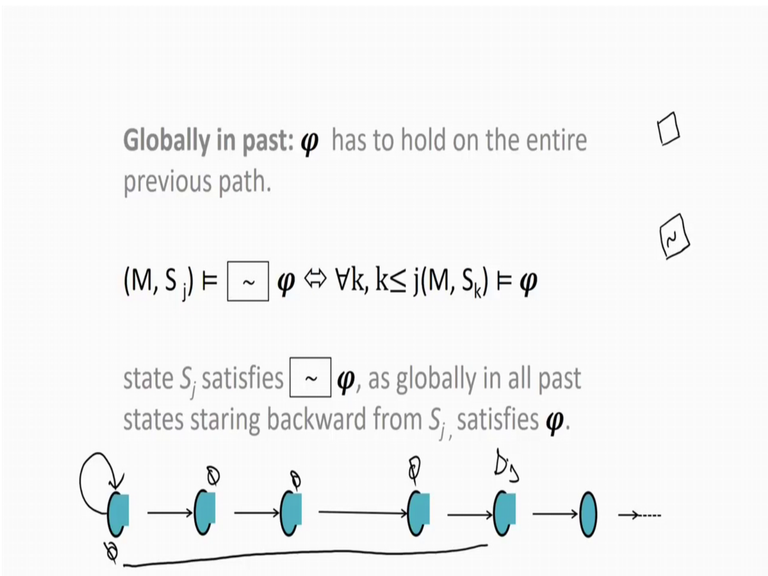
So, if this is the next operator then with tilde. Now, we are going to get the previous operator. So, this is says that as a satisfied that previous phi as it previous state S j minus 1 satisfies phi it is M. So, if I say that S j is the state and S j minus 1 in the previous say state phi holds then we said that in this particular state previous phi holds, ok. So, this is the notion of the operator previous which is corresponds to your next operator in future logic.

(Refer Slide Time: 45:22)



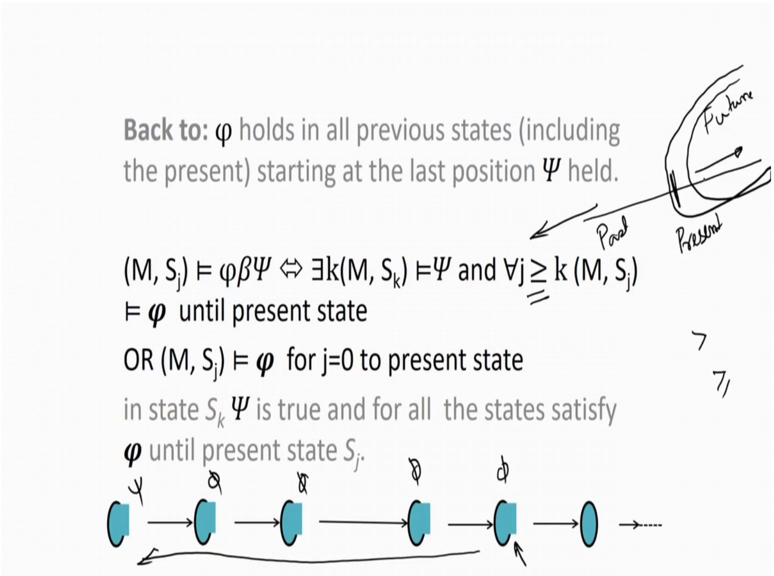
Similarly, second one is your eventually in past. So, in future temporal logic we have the future operator which is given by diamond. So, here using the same diamond symbol and with the tilde inside it, ok. So, this is the eventually in past. So, it is also same notions that we are having if whether it satisfies eventually in past phi. So, if we are in a particular state say S j and we are going to look for some previous step S k, where phi holds and we say that in this particular state S j eventually in past phi holds, ok. So, this is corresponds to the future operator.

(Refer Slide Time: 46:11)



Similarly, globally in past, so that box is used for the global operator. So, box and inside that if we put a tilde symbol and it is the globally in past. So, what basically the notion is same if we are having this particular state say S j if all the previous state say we are starting this is starting state in all the previous state say phi holds then we say that in this particular state S j globally phi holds, ok.

(Refer Slide Time: 46:56)



Similarly, we are going to have this particular back to; so phi holds back to psi holds so that means, if in a particular state say phi holds and all the previous state phi holds still we get a state where psi holds then we say that phi back to psi holds in this particular state if it is a state formula or if it is a path formula then we will say that in this particular path phi back to psi holds, ok. So, these are the future operator.

Now, I think I have also not mention properly in this particular case in most of the cases you said I am talking about j is greater than equal to k, this is greater than equal to k, we are not simply writing it is greater than, but we are writing greater than equal to k. Similarly here also I am writing less than equal to k. So, this is basically in case of your past logic or future logic also you will find that we are going to have that notion also say less than equal to here also greater than equal to k. So, in those particular case it is have been said greater than equal to k. So, we are talking about either greater than equal to object.

Now, just simply look into this particular scenario I am going to explain over here. If S j and we are going to get some k such that phi is holds in this particular k, ok. Now, what will happen if here I am writing that k is greater than equal to j, ok? So, this is the time S j, j plus 1 like that we are coming to some step S k where time is progressing in this direction. So, if it is having k is greater than j. Now, what will happen if k equal to j? That means, if this is k equal to j means this particular state itself is your S j also, ok. So, if k equal to j I can say that this is equal to. Just say in this particular case now I am coming with a simple model say this is a S j and here said that in j itself phi holds, ok. So, this is the scenario I am just looking into one particular state S j in this particular state S j phi holds.

Now, if I am going to ask whether M S j in the state S j of the model M whether it models F phi or not or whether F phi to in S j are not in this particular model just consider this particular model. Now, what is your view in this particular case? Whether this state S j whether it models F phi whether F phi feature of phi is true in this particular state. So, if I look into the definition formal definition then what you will find that, since here I am saying that k is greater than equal to j so that means, I am coming to this particular scenario that means, in the state j itself phi is holds or phi is true then I can say that in the state S j F phi is also true. So, if I say that whether this model.

So, according to our definition here I can say that it is true our F phi is true in the state S j; because why I am concluding like that? Because due to this greater than equal to j. So, if k is equal to j also in this particular case I can say that yes, future of phi holds in that particular state S j. So, what is the minute observation over here? That means, if I am in a particular state and phi is true and in next clock pulse basically what will happen, already I say the digital system works on clock pulse. So, if I am having in this particular state S j the next clock pulse as per the transition will go to S j plus 1, ok.

So, now, what we are saying according to the definition if said that if F phi is true in S j then F phi is also who in S j. Why we are concluding it? Because, this is as per the definition of this future operator; so, in this particular what we said that we are going to capture a future behaviour, but future is included in the present state itself, ok. So, in this particular case I can say that the future behaviour is included in the present behavior itself, but if I am going to write say instead of k is greater than equal to j if we are going to write at k is greater than j in that particular case what we are doing we are excluding the future behaviour from the present state.

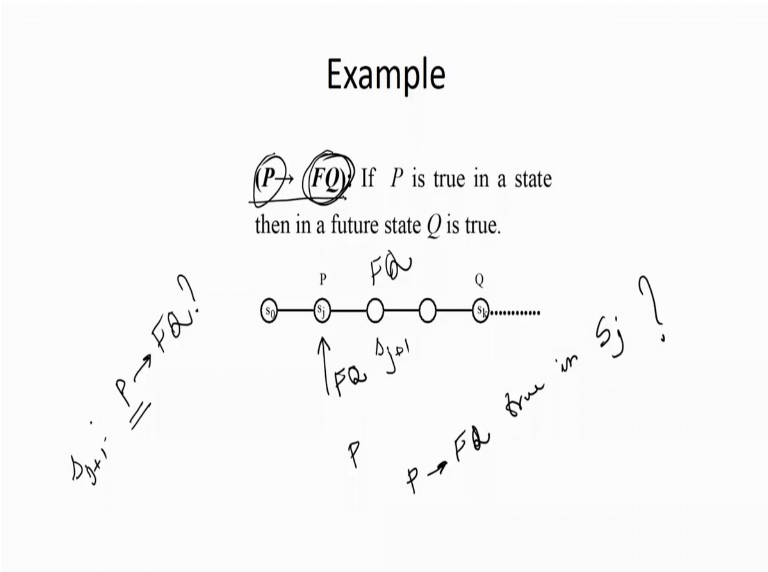
So, in that particular case if I say that k is greater than j in that particular case in this particular state S j F phi will not be true. So, whatever semantics or meaning we are giving over here basically future includes the present also. So, when I talk about a future behaviour that means, presence is also included in the future behaviour, so that is why F phi future phi is true in S j if we having such type of behaviour. So, in a in next state say phi is false still F phi true in S j because future includes the present. So, this is the notion of timing that we have incorporated while giving the meaning of those particular temporal operator.

We can define the meaning of the temporal operator where we are going to exclude the present behaviour from the future behaviour then those definition will be slightly different because instead of greater than equal to k it will be only greater than k. So, in all those particular cases what will happen? We are saying it is greater than equal to k that means, future in include stop present behaviour also, but if we want to exclude it then here we applied greater than k. So, we have to be very careful while using those particular temporal operator. And the meaning truth values of the temporal operator will depends on the semantics that we are going to use. So, in our case basically we use this particular meaning that we will write k is greater than equal to j that means, we will say that future includes the present behaviour also otherwise the truth values will have some different effect, ok.

So, these are the so similarly in case of future also or in past logic also that similarly that less than equal to k is coming to (Refer Time: 54:40) that means, past is include in the present, presents include the past also, ok. So that means, what basically we can conclude here? Here I can say that if this is the time line and if I said that this is my present and then here I am having future and here I am having past. Now, how we are defining a semantic future includes the present that means, this is the future it includes the present also. But if we want to exclude the present from the future then what will happen we have to start from somewhere here, ok.

So, accordingly the truth values will be different in the present step. In this, whatever you have defined over here we have defined with respect to this notion that future includes the present, ok. This is the basic operators that we have in temporal logic and temporal logic is divided can be categorized into two part, one is your future temporal logic and second one is your past temporal logic. With the help of future temporal logic we can reason about the system behaviour in a future of what a future how system is going to behave in futures, but if you want to know about its past execution past behaviour then you can use the past temporal logic. Those are, after discussing of those particular temporal operators just see some example how we are going to interpret it.

(Refer Slide Time: 56:20)



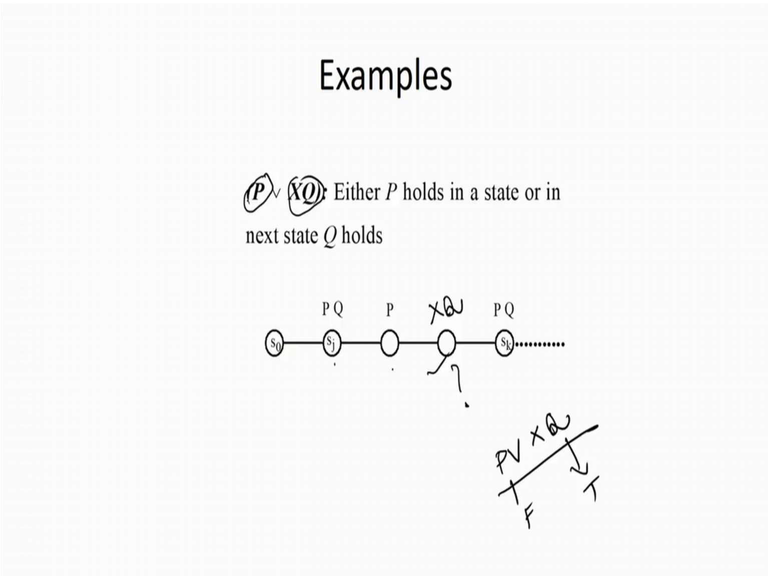
Here I am giving one example say P implies F of Q it is a temporal formula because a Q is a formula and you can say that it is some propositional formula, then FQ is also temporal formula future Q, P implies FQ is also temporal formula because I said that any logical connective can be used in the temporal logic. So, it says that if P is true in a state then in a future state Q is true, ok.

So, I am going to talk about the state say if I am going to consider this particular state S j then what we are going to setup? P is true at that particular point and if I look into these things (Refer Time: 57:13). If from S j you are starting in this particular way somewhere in future we are going to get S k case that where Q is true. So, in this particular state I can say that FQ true. So, when I am going to look for this particular P implies FQ. Now, whether this is true in S j if P then FQ, so P is also true and FQ is also true that means, in S j P implies FQ is true because in future we are going to get some state where P true.

Now, the next state is a S j plus 1, whether in S j plus 1 whether P implies FQ true. Now, if I look into it in this particular step from here in future somewhere I am getting S state S k, where Q is true. So, here FQ is true, ok. Now, here is a P is not marked over here that means, P is false over here. So, whether or not P implies FQ is true at the particular state S j plus 1. Now, is just look into the notion of that implication operation, if P is true then Q must be true this is the P implies Q if P is false then whatever may be the truth values of your Q P implies Q is true.

So, since P is false at the particular point. So, false implies anything is true that means, P implies FQ is also true in the state S j plus 1. So, like that we can see the truth values or the given temporal logic formula after knowing the truth values of which component. So, in this particular formula I am having two component one is P which is the atomic proposition it can (Refer Time: 59:29) true and false and FQ is another temporal formula. So, P implies FQ is a temporal formula.

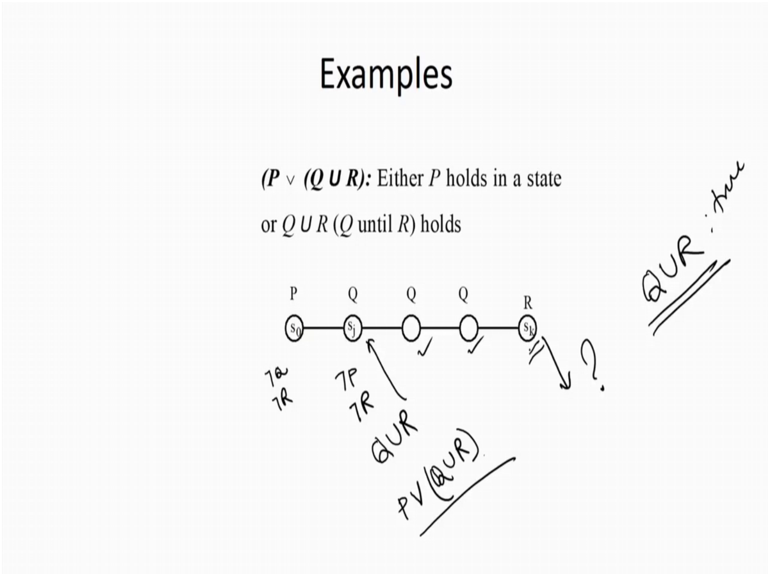
(Refer Slide Time: 59:36)



Now, it says that P or XQ either P holds in a state or in next state Q holds so that means, if P is true in a particular state then P or XQ is true. So, since P is true at the particular point, so P or XQ is true; since P is true at the particular point so P or XQ is true. What about in this particular state? P is false, since P is false so our second component need to be true. So, in this particular say in the S k Q is true.

So, in this particular state that means, XQ is true because in next step Q is true. So, XQ will be true over here, so P or XQ. If I look into it then in this particular state P is false but XQ is true. So, P or S Q is true at that particular state, ok. So, again it is having two component XQ and P and they are this (Refer Time: 60:46) over here.

(Refer Slide Time: 60:48)



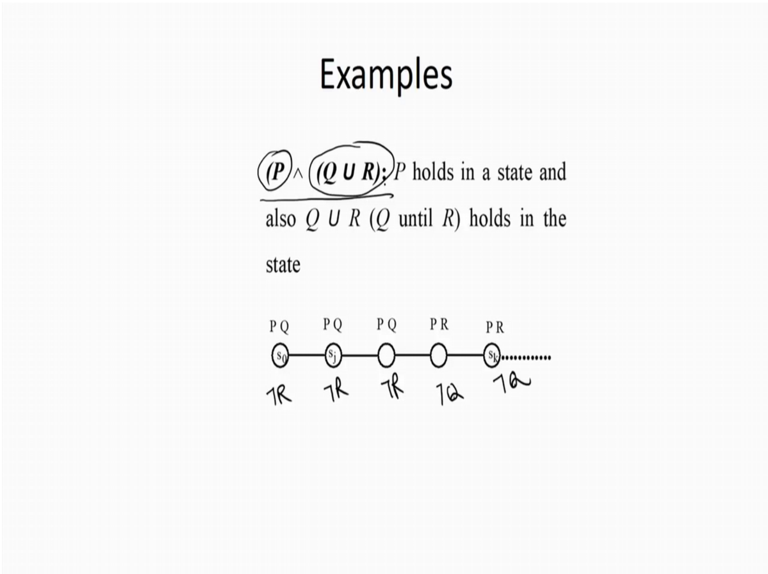
Now, whatever the next formula P or Q until R; so, whatever that I am writing P Q R in you just think that these are the atomic proposition they are having that truth values either true or false if it is marked with P that means, it says that s 0 P is true, S j Q is true, but it is not marked with P and R you can say that this is your not of P and not of R. So, what does it means? Q is true in S j, but P is false and R is false similarly if I look into s 0 P is true but Q is false and R is false over here.

Now, we are talking about P or Q until R. Now, whether in s 0 whether it is true because P is true itself so it is (Refer Time: 61:42) of the other component. So, P or Q until R is true over here. Now, when I talk about these particular things S j P is false at that particular point. So, Q until R must be true over here. Now, you just see the execution that Q remains true until R becomes true; so here Q until R is true. So, in that particular case P or Q until R is true in the particular state.

So, similarly you can say that in this two state also P or Q until R is true. What will happen to this particular state? S k, now, P is false. Now, you have to see what is the other component Q until R, as far our notion of timing we said that future in (Refer

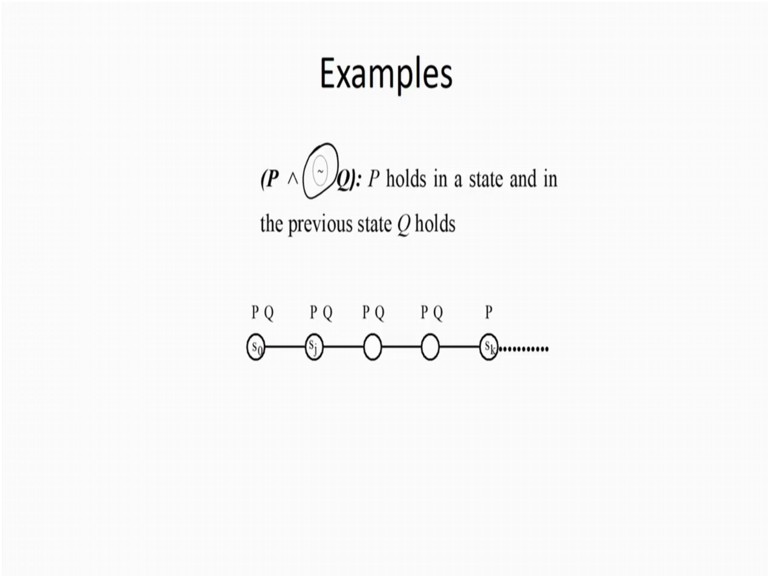
Time: 62:43) present state since R is true over here. So, in this particular state P until Q is also true. So, P is false, but Q until R is true. So, P or Q until R is true in this particular state S k. So, this is important you have to see how we have captured time. Here we said that as far our semantic future (Refer Time: 63:08) the present behaviour if we are going to exclude it then this formula will be false in state S k, ok. So, the notion is like that whether future includes the present or future exclude presents. The semantics that we have defined over here is basically future includes the present behaviour also.

(Refer Slide Time: 63:30)



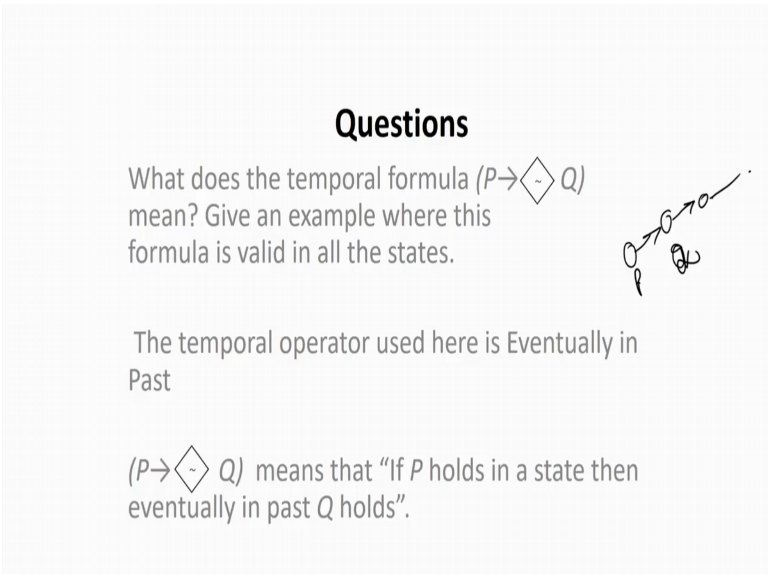
Now, if I am going to have that this is the similar way. Now, you can see whether P and Q until R. So, Q until R one temporal logic formula, P is another one, so now, you have sense the level. So, what it says in this 3 step? P and Q are true, but R is false in these 3 step. Here P and R is true, but Q is false over here. So, here Q is false. Now, you try to find out the steps where P and Q until R is true and you try to find out what are the states that this is false.

(Refer Slide Time: 64:08)



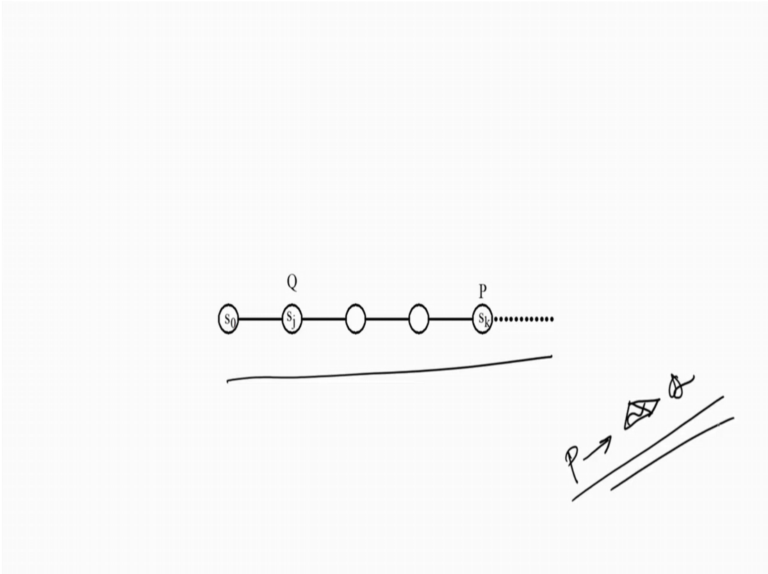
So, similarly this is another example. Now, you try to analyse it. Here I am using the past temporal operator where we are saying it is the previous, ok. Now, you try to look for those particular state and say what is the truth values of this formula.

(Refer Slide Time: 64:24)



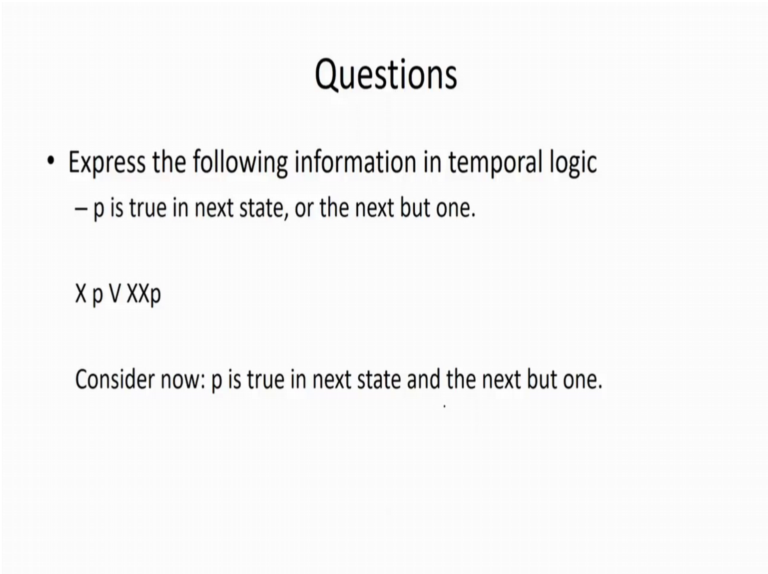
Similarly, another question we are talking about what does the temporal formula P implies that Q, diamond inside that Q that means, this is basically not previous basically back eventually in past. So, P holds in a state then eventually in past Q holds. So, it is in past basically. Now, you try to give a define some model like that whatever we are talking about and level them with P and Q and try to find out the states where this particular formula is true.

(Refer Slide Time: 65:04)



So, maybe you can use this particular model and label it and see whether the formula you are talking about P implies Q, this formula where it is true and where it is false.

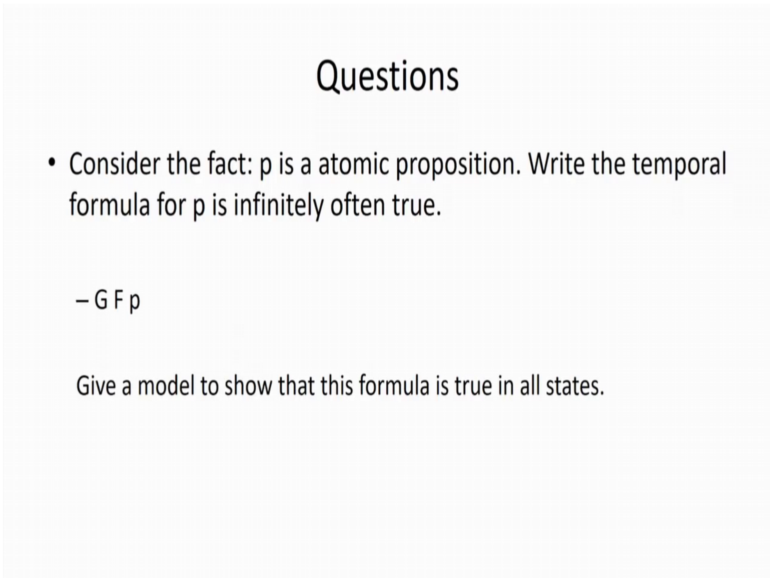
(Refer Slide Time: 65:21)



Now, another question I am going express the following information in temporal logic p is true in next state or the next but one, that means, it is talking about either it will be true in this particular state, this is s 0, s 1, or s 2. So, this is the behaviour that means, either p may be true over here or p may be true at the particular part. So, how I am going to describe this behaviour in temporal logic? So, it says that in next step p holds or next to next step p holds, ok.

So that means, this temporal operator can be again used. So, Xp is a temporal formula after knowing that truth value of Xp I can say XXp, ok. So, this is a temporal logic formula. So, p is true in the next step and the next part one. So, this is the things.

(Refer Slide Time: 66:23)



Consider fact p is an atomic proposition write the temporal formula for p is infinitely often true that means, wherever you go in future p is going to have a going to be true, ok, infinitely often. So, how I am going to capture it? Say this is infinitely often say in future (Refer Time: 66:48) from wherever you go in future p is going to be true and this property must hold globally, ok, so that is a way G F p it says p is infinitely often true. So, globally this property must hold wherever you go in future p must be true.

So, like that if we are going to have some notion some specification where the timing is involved. So, such things can be captured with the help of this temporal logic but remember all cannot be captured because we are having also restriction of this particular temporal logic operator. So, for that we may have to go for more operators. So, basically talks about the expressive power of the temporal logic. So, we are having several notion or several behaviour, some of the behaviour can be captured with the help of the temporal operators that we have discussed today, ok. With this I am going to wind up today. So, in next class we are going to continue from this particular point.

Thank you very much.