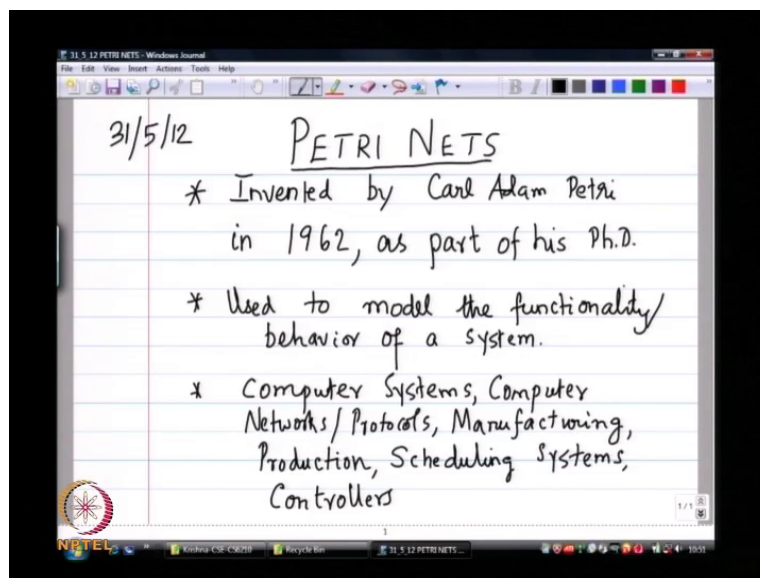


Performance Evaluation of Computer Systems
Prof. Krishna MoorthySivalingam
Department of Computer Science and Engineering
Indian institute of Technology, Madras
Lecture No. # 39
PetriNets-I

This topic is on PetriNets, we have look at different techniques for mathematical modeling of systems right, we have seen in the basic queuing networks ora queuing system, (()) the mm 1 queues is an marks out chains, you have looked at close queuing net works MBA and the related techniques. So, petri nets is a other form of mathematical modeling to formally specified a particular system right, normally specification we can also derived some performance results also from this petri nets modeling.

So, it is very similar to mark out chains in terms of applications right; we can essentially model the behavior of a system, and then try to derive some properties from that particular model.

(Refer Slide Time: 00:55)



So, the name petri nets arises, because this was invented by I guess Carl Adam Petri, I guess we can see invented, because he came up with this and if you look at Wikipedia forthis page

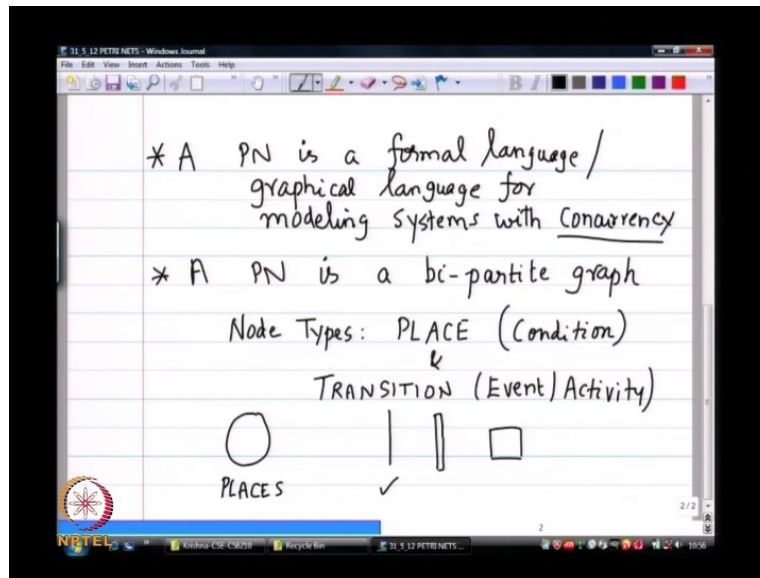
on petri, he written that one page on himself in says actually discover, he came of this in 19, when he was about 13 years old the 1930s, but then he formally here it as a presentation in 1962 as part of as Ph.D theses.

So, we did not know about how true it is, whether he came up with this 13th year are not, we will simply use right and 1962 as the age 1, as the time and this was actually formally announce to the world right, this is the part of this Ph.D theses. The Ph.D theses itself was publish in 1966, but the ideas were originally conceived in 1962. So, this as be said right use to model the behavior of a system right; model the functionality slash behavior of a system. And what the system is we will see examples of the systems as we go will on. And if you in terms of applications right, there are lot of domains where this can be applied one of course we know, we know computer systems where we can actually model this computer communication, networks we can model right.

So, we can look at computer systems, the word thus the type that we saw before, that we model with MBA and so on, are computer communication network right, so computer networks protocols and related fields, but in addition it is got lot of other applications in like, manufacturing for it is all right, if you look at some other examples you will very clearly see that you know, just like any computer processing system can also be map to manufacturing system, where there are parts, there are components and you are do some processing on those components and then write combine those parts in two different system.

So, manufacturing systems, production systems were they actually manufacture part in terms in to product system in and also sophisticated scheduling systems right, this is scheduling not from merely our computer systems, networks from a scheduling, but also scheduling in terms of jobs in shops right; the machines available and jobs to the processed on the machines and we need some sort of scheduling systems, there also controllers right, so we can look at controllers as in your PLC controllers, that we see right embedded controllers, let we see on various devices, other program logic control is right. So, this can also be model behavior of those can be model with the help of petri nets, that is them basic, some of the applications that you can see with that that, can be del with using petri nets, so that is the first part of who came up with it and what it can be potentially used for.

(Refer Slide Time: 04:19)



So, now, let us look at what is formal definition right, so petri net we want to use PN to represent the Petri Net right, so petri net is a formal language it is formal language to represent the system; it is also a graphical language. So, by looking at the system graph, just like state machine like right, state machine is also a graphical means of representing system behavior, this is also a graphical language; and this is especially for modeling systems with concurrency this is what does key right, how do you deal with concurrent systems model.

And if you go back to the 1960s computer science in concurrency at the time probably just, people just talking about right, you do not use concurrency as we use today multi thread system are we common or parallel system is become quite common, and so this is systems where there are concurrent operations going on; concurrent processing there are going on that communicate with each other and you want understand if, because of these concurrent processing right for example, we will see example in terms of right, the basic threading and things like that, so but it is, that is the basic idea right, where are the concurrency you can try to use this petri net system to model system behavior and from that derives some sort of performance metrics.

So, formally so petri net, if you were to talk about in the graphical, or in a graph theory term this is bi-partite graph, so it is a bi-partite graph containing two types of nodes and then you have graphs connecting those two types of nodes. So, two types of nodes are the nodes types are vertex types call PLACES. So, place is one type, places whatever is place where you

store something, store token for example, think of it is a box or buffer something that wholes, so that is called a place; and is this can also be used to think and terms of system state right, so this represents some sort of condition that has to be satisfied right in the communication, in the process that we are trying to model.

So, place is term that going to used and the next term is transition, so transition is the other node type, in this bi-partite graph and transition essentially represent some event that occurs are some processing activity that takes place. So, there are items held in places and all those items when some conditions are satisfied at transition will trigger, it transition will occur and that transition is name implies will changes state from, change a system from one state to some other state, so that is transition. As the transition like is it, corresponds to some sort of activity, so for example, if you look at say use use the program is waiting for data right, waiting for some input from the keyboard.

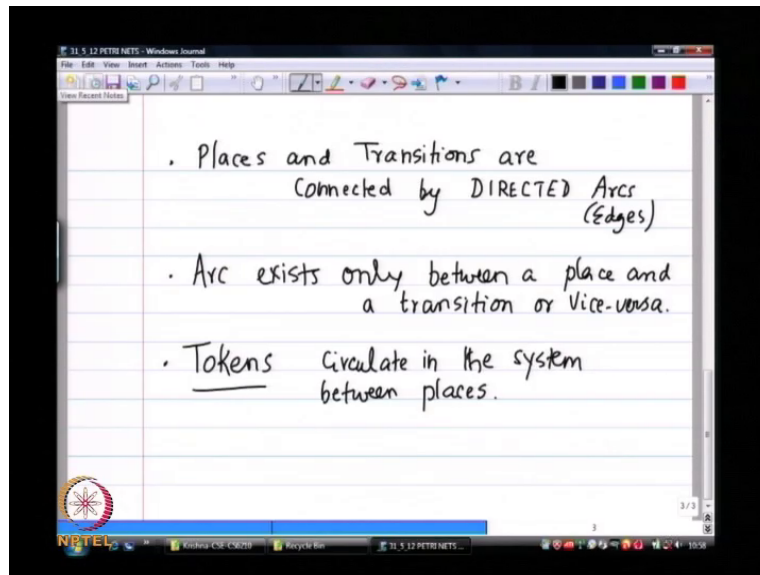
So, as in is the keyboard input arrives it gets stored in a keyboard buffer right or in the system cornel buffer corresponding to the keyboard, and then you have some logic that works on that character that are character rate from the keyboard. So, that processing is the activity that is the transition that takes your input keyboard values and then converts that to some other value right.

So, for example, user types in excess input, the processing is a output esquire, so the logic is represented by the transition and then intern produces output is esquire, so that goes to some other place, so that is these are the main components, place and transition. And we use circle to denote PLACES and the transition there are different ways, we normally simply draw straight line to represent transition are sometimes it will rectangle right, thin rectangle like that represent transition and there are differences were are sometimes just people use even boxes to represent transition. We are going to use this straight line to represent transition, the places are simply circles and inside this place inside this box as you can find tokens right, that is the other component of the system.

So, that are places and transitions and there are tokens that circulate between this places in the system by are this transitions, what transitions like gates and it goes from place to place through this different gates and the gate can be open only when certain conditions are met, that is the basic idea otherwise the gate is kept closed right, so in the gate opens the horse jump from one part of the field the other part of the field right, so that is the basic idea. So,

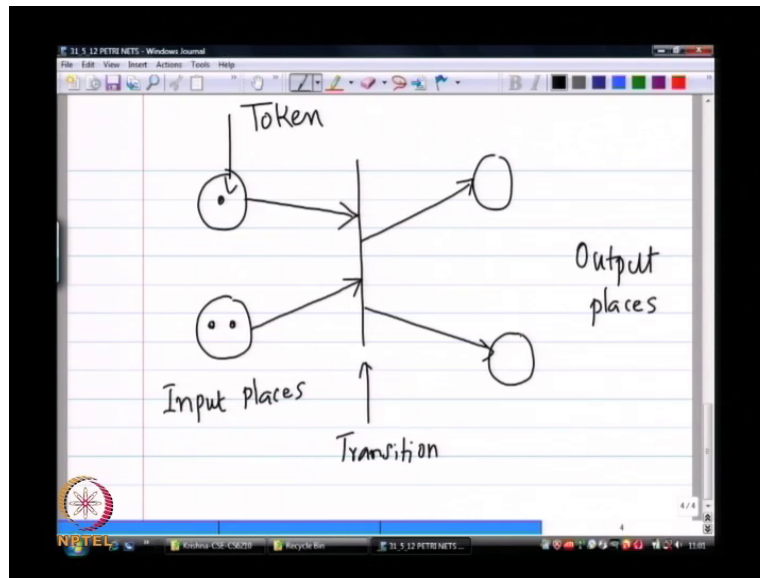
now these are the three node types in our bi-partite graph, so what will be need in the bi-partite graph the edges right, the arcs the connect these to node types.

(Refer Slide Time: 09:23)



So, we have now, so the places and transitions are connected by directed edges are we also called them arcs right, so directed bi-partite graph. So, now let us, so they arccan exist only between a place in a transition viceversa, but you cannot have an arc between two place that is why it is a bi-partite graph right otherwise it is so. So, the Arc exists only between a place and transition (No audio from 10:07 to 10:17) or vice-versa. So, that these are the,that is it this basically defines also is notion of and then we have of course tokens right,so tokens circulate in the system between places (No audio from 10:39 to 10:51), so these are the basic entities that are available in the system. So, nextlet us look at the simple example right, thatfigure as to show what is going on.

(Refer Slide Time: 11:08)



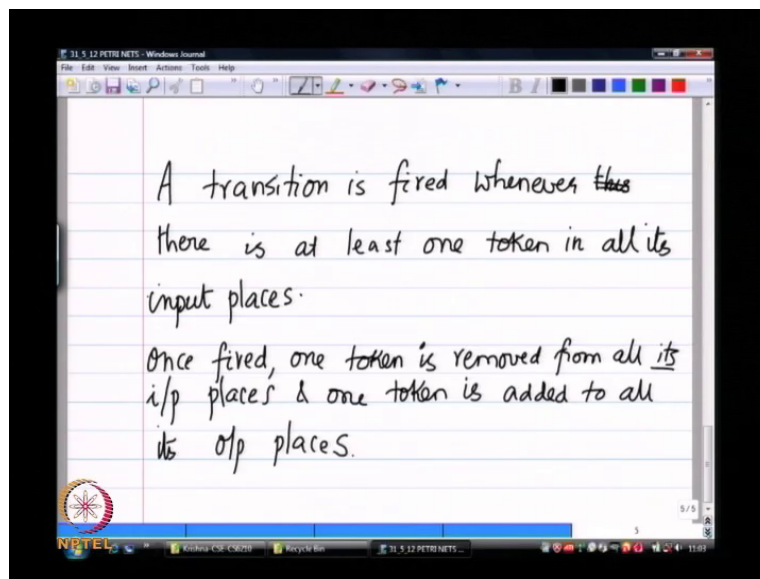
So, here is a place, here is in other place there is, this place is one token, this place has two tokens, here is a transitions and we have state in there, there is in arc connecting this place to this transition and this place to this transition; and then from this transition you can have output transition that go to other places, so this is your basic model of the system components. So, these are on the left are input places with respect to this particular transition, right this input place could be output place for some other transition, for this given transition this is the input place, and then these are your output places and this is the token in the system.

So, tokens get created in these places and they get moved on the system, next we go will on, now when does the transition takes place right, so this simplest rule for the transition to take places that as long as there is, if there is a multiple input places for a given transition, if there is one token at present in, at least one token present in all the input places that are connecting to a transition, then the transition will fired. And as the result of the transition what will happen is, one token will be moved from will be removed from here, one token will be removed from here, and then you will be placing one token here and one token there.

And the number of inputs, number of outputs need not be equal, I can have less number of input places and more number of output places viceversa, all we are saying is when a transition occurs I delete tokens from all the input places and add one, one token to all the output places that is the very basic definition of what happens, we can make this more

complex later on by doing multiple tokens in things like that, but simply speaking that is why transition occurs, whenever there is an input in this basically all these conditions have to be true that is what I said right, when all these conditions are true the transitions fire as, as the result of the transitions some values get deposited in these corresponding output places. So, implicitly there is copying and things that are going on, but as well as system goes token is going to be created in each of those output places is that clear, questions no, so will write it down now.

(Refer Slide Time: 13:44)

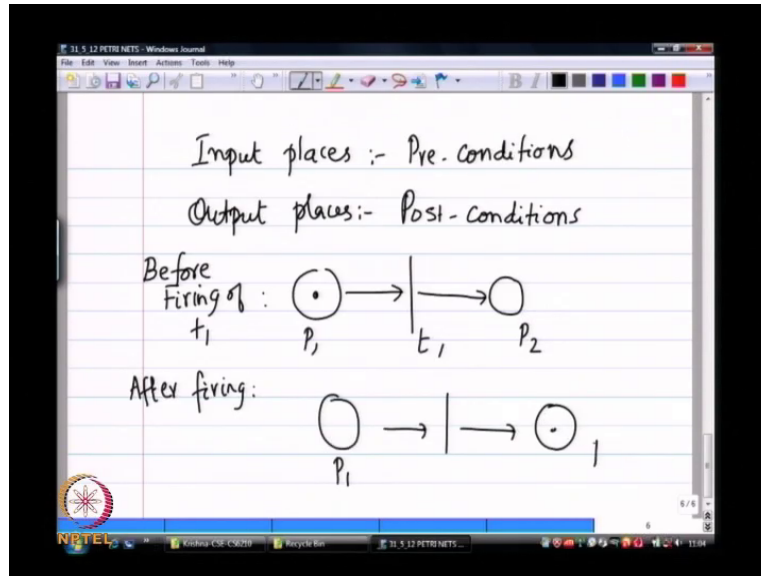


So, transition triggers in the basic definition, we said triggered or fired right, whenever there is (No audio from 14:52 to 14:00) at least one token in all the input places, in all its input places that is point, and once fired one token is removed from all the input places and one token is added in or added to all output places, I would say all its input places is added to all of its output places that is the fundamental think of a transition. Now, the lots of hidden details right when is transition, how long it is transition take to fire is it instantaneous right, when a transition fired immediately does a packet get delivered, all is there delay right processing delay, if you look at a transition is event or in activity naturally it cannot be instantaneous right sometimes it can be, sometimes it need not be.

So, all those other extension will come later on, but that is the basic notion of ordered transition is for this system is cancel. So, will set of go through some example, so in other

word all the input, in other says in input arcs represent conditions right that are to be satisfied for this event are transition to be activated, that what is this basic definition.

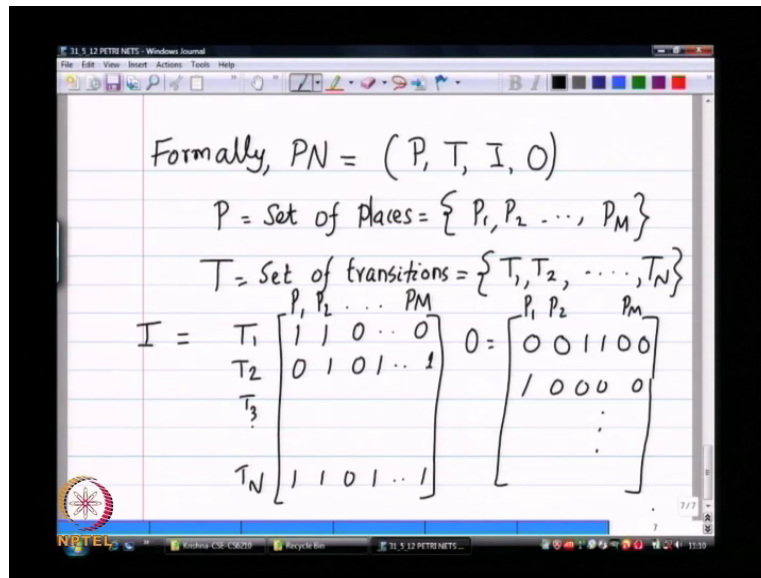
(Refer Slide Time: 15:54)



So, if you can interrupt the input places as pre-conditions, if you remember when you write this in c program, in programs in general right, we talk about pre-conditions. So, pre-conditions when they are satisfied right only then this function can be executed right, so they represent the preconditions and the output places represent the post-conditions this is in terms of our programming terminology right. So, post-conditions could be right number of items in the buffer is x plus 1, before it was x , after that it was x plus 1 those of the pre post-conditions, so for example, if I want to say right this is simple system right, so this is the input place, that is the output place. So, this is before firing, let see this is t_1 right, this is p_1 and this is p_2 , so before firing of t_1 this is about the system state was, then after firing system state will find this empty and this will have 1 buffer, so p_2 will have token, p_1 will have a no token; in case p_1 have more tokens two tokens in that case p_1 will still have one more token left, but p_2 will have this one extra token add it right, so that is the basic right.

So, by default one token is removed and one token is added to the corresponding from to thus in input and output places, so that is the, so formally if I want to represent this right, so this is a graph, so they any questions on this basic representation so for right, so in activation happen one token is remove from all the input, one token is added to all the output places.

(Refer Slide Time: 17:57)



So, begin to state this formally right, a petri net called PN is defined by four triple, there is P, there is T, there is I, there is O, these are the components of the petri net. So, what is P, P is the set of places and that is represented by P 1, P 2 up to P M right, so M is the cardinality of places, so that M places in the system, T is the set of transitions and there are N transition in the system, so we will call them T 1 to T N.

So, what is this INO, any ideawhat I could be

(0)

All the

(0)

All the input places right, so what would be the, what would I B

(0)

I am sorry, it will be the set of input places know for every transition right, I could have one or more of the input places connected right and I have in such transitions, so there are M places and N transition. So, what could be this I B matrix right, matrix ofwhatever M by N or N by M, so this I is a matrix that basically tells you, which are the different places, input places that are associated with the corresponding transition; and entries will be 1 or 0 in the

basic case, we can have more sophisticated cases later on, based on multiplicity of the arcs and things like that, but basically this is just you have connectivity matrix right, this is a set of gates in the system, because it is a directed graph I need to provide one this way and one for the other way, so these are all in the inputs right.

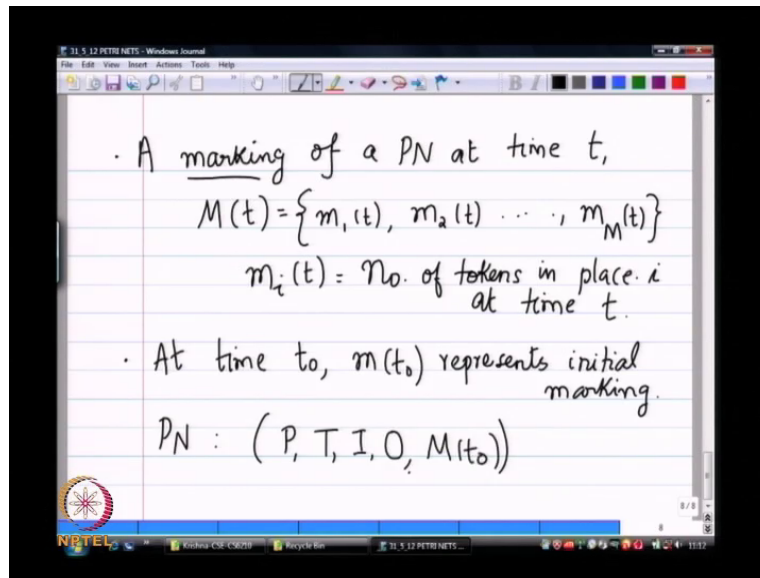
So, for example, I am it say that P 1 and P 2 are inputs to transition 1 and may be P 2 and P 4 and may be P M are inputs transition 2 and so on right. So, some sort of structure is the, this is the input, this you definition you have to come up this you given a problem, given a system you model the system and then part of the end of the modeling will come up this particular matrix.

And likewise, you have the O matrix also which will specify for each transition, what are the corresponding, so same thing the rows of the same and the columns of the same, so it is possible that of the end of transition right, you will have and if you look at it there is technically no loop at right, this are loop less, you do not want to go from input place, the input and output places are it can have loops they will allowed, but now for let us assumed that there are this is a loop less system right, where they output places are different from the input places, so like this could be anything right, this could be just 1 0 0 0 and things like that, so these are a specification for the network itself.

So, this is now a static definition of the system, of this network with only this definition what can I do, you can do anything, we need to have something this is, we need something dynamic in the system right, what is the dynamic part where is the dynamic it is come in, when you bring its tokens. So, now this system state is captured by the number of tokens that are present in all those different M places right.

So, there are right M places, like if you look at this CPU disk problem there is 1 CPU queue and there are 4 disks disks queues right. So, the systems state is captured by the number of jobs in each of those 4 queues, each of each of those queues right, each of 5 queues in total that is the system state right. So, therefore, this static definition of a system is this, but you lot also bring in the dynamic definition, so will have to have one more tuple to this, so that is what is called us the marking.

(Refer Slide Time: 22:29)



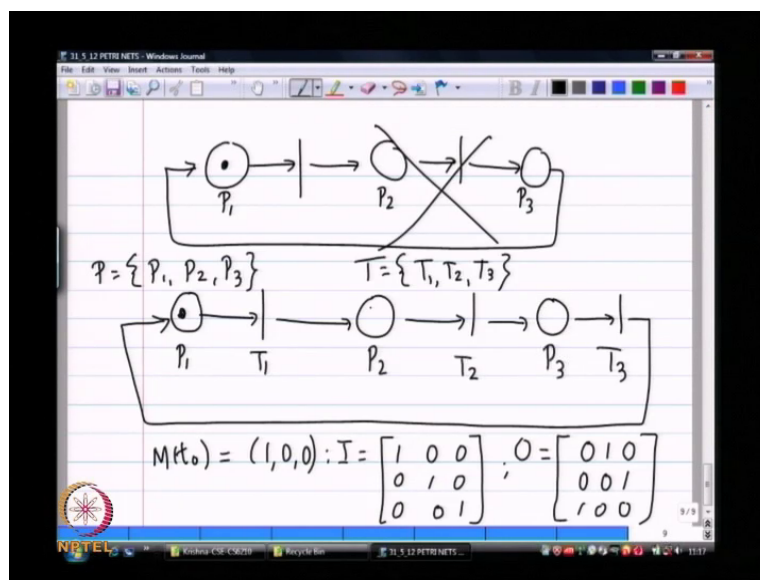
So, will define a marking, marking of a petri net at time t , we will associate know, because this is technically moving through time right, so at timethe marking of a system is simply given by the number of tokens present in each of those various places. So, where $m_i(t)$ is the number of tokens in place i at time t , this is the definition of marking of the system, so this it present the current active state of the system rightdynamic state of the dynamic state of the system itself right and then we used this show, when we start just like in the case of state transitionmark out chain rightor state transition diagram you always what is an initial state this right. Let us start somewhere, there is big chain, big graph there different states of connected to different ways you always say this is the initial state right, so in addition to set ofthe state transition matrix you also specify the starting state, so likewise you all to have to, also have to provide the initial marking of the system right.

So, sometimes t_0 which is the initial time right $m(t_0)$ represents the initial marking, there is the little bit of confusion here, because m is appearing both as the number of places as well as the marking of the system right. So, just bar with that we know one marking with that m , when I use the term m in terms for the number of places we need talking about the other m , so this is basically, so now we are the petri net definition will now actually include you are set of places, set of transitions, inputs matrix, output matrix as well as $m(t_0)$. So, now, if I give this to you saying this is where this is my starting state of the system, so now we will start with may be one token in one of those places are nine tokens in the case of CPU disk

example right, add put nine tokens somewhere, if you remember MBA, we did the same think right, how do you find out the number of jobs in each of those queues, we start wise simply saying that all jobs end are equally divided amount all those m queues, so it is n by m is starting state they need tried tried and find out the steady state in terms of number of distributed, like ways will start with putting all the nine jobs in the CPU queue and then see how the system progresses with time and finally will come to a steady state system, say that could be m of t 0 right, so that is your initial marking of the system.

So, from this initial marking then what will happen is, we can fired transition that are various transition that there are possible, because these are all transition in parallel right, the different combination of transition it can take place and then the state of the system the marking of the petri net keeps changing with time and that is something that we will now try to study and go all that is main think right. So, given the petri net given the initial marking, what is the sequence of markings that the system will go threw, will the system lead to dead luck right are if does not lead to dead luck what is the probability of the system being in the given mark states, at given point in time, all those thinks will what you want you want to find out, and those will translate to our performance petri net. So, let us look at few more examples, and some sort of you know classifications of the petri net components, so questions no.

(Refer Slide Time: 26:31)



So, let us look at one petri net right, which is basically a system which is you know assume that is the machine right that can be loaded and that will do some processing and then after

that it will do some one loading, we can think of this is a machine are as a computer computer science terms, this is just as CPU that is processing something. So, the here is data is available it gets process gets (()) in some post processed bufferwere it can be take in for printing are something like that, so and thenremove this is also loop in this particular system. So, we havethree states sorry three places P 1, P 2, P 3 and then there are two transitions sorry, I missing the transitions.

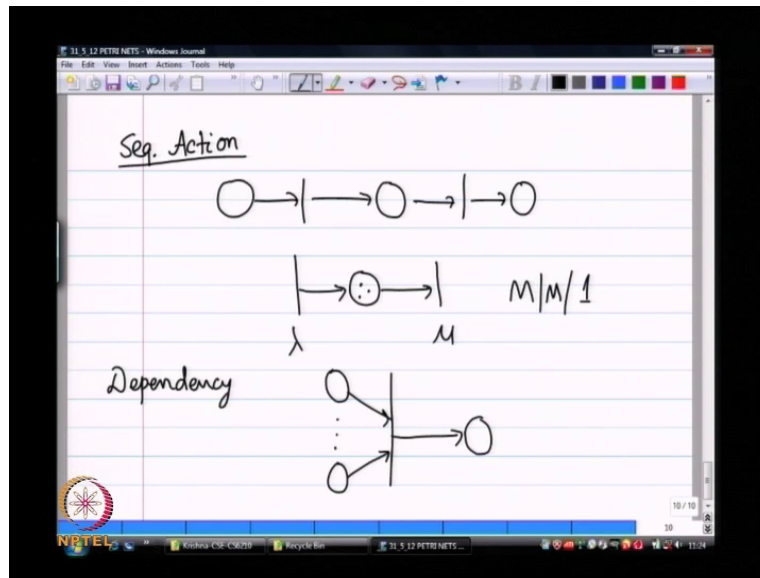
Let us, vacant we need three transitions right, because from I cannot go from transition to transition right; we will go back draw this again, there is P 1, T 1 that is one, we go to P 2that is a next storage place and then that goes to the next transition just T 2, then it goes to P 3, then it goes to (()) transition T 3 which then finally loops back to, so this is the static definition of the system, these are all that notes and corresponding arcs that is defined. So, now an a bring let us an initialstate is 1, 0, 0, so the initial state M of t 0 is equal to 1, 0, 0, so what that is the initial state.

Now, the input matrix willsorry, I matrix will be what,what will be I matrix look like, remember the I matrix rows are all transitions right, so T 1 what are the input places only P 1right, so only P 1 others are 0, 0, T 2, 0 1 0, 0 0 1right, this is are standard graph representation right, because when I have impost input was the system I should specified that right. So, think about did in the programming right, if I have to give you this model this will way of inputs to you this is representation of the petri net system right, I will tell you the set of places and transition and likewise O is what,what is the output 0 1 0, 0 0 1, 1 0 0that is all, right this now completely specifies may input of the petri net as such.

So, P is the set of, so if you on to be more complete P is the set of states P 1,P 2,P 3 that is input, then T is T 1, T 2, T 3 this is also given to us. So, now, the representation as well as the initial value, we have to look for some properties of that petri net, because we ultimately our goal is to get some set of performance results from this system right, I think a did something and suppose to alright,is that clear, questions.

Now, there are different you know types of I guess sub components of the system right, so we just talk about some of those things in sequence.

(Refer Slide Time: 30:13)



So, for example, you could have a sequential action, this is an very straight forward where there is you know place that feeds a transition which inferred feeds like, this has a sequential action right data comes in which processed here, get processed here and finally, gets in store there and its possible let you can have a system with a transition with no input, that is somethingwe are not yet talked about, but I can have a system like this, I can even have a system like that where the first transition has no input at all, which means it willit will keep on triggering right.

So, it is just like a while loop at just keeps triggering only think is a add some delay to this, if each transitions take some time say 5 millisecondsare exponential time with some meant 10 milliseconds, then this will that is the processing time for the particular system right and then it gets inputs. So, whenever it generates something it puts the token here, and then whenever there is thetoken it will be simply picked up by this transition as long as that is one token this transition will fired, and after it process is it there is nothing for it output in the system, this is also very simple sequential action.

And you know what this kind ofpercent, if I tell you that the times spent each of those transitions right is exponential with parameter lambda and parameter mew what could this be, what could this be, what could this system be, any ideas.

((0))

M/M/1 queue right, because this is exponential time, this is your packet arrival process right. So, everyone exponential right, every one over lambda with me, one over lambda you generate the packet, when a packet comes it goes here when whenever this queue is not empty this is a server, this a arrival process, this is a server process right and whenever picks packet for processing, customer for processing it spends one over mu to be mean one over mu time on that process and then that is it, there is no output you do not really care about it right this is your basic M/M/1 queue, if you look at it where there is no input and where there is no output, so this is actually representing an M/M/1 queue system.

Now, the question is what do we do with this, how can with, how do you use this for performance evaluation I will come to that later; we will look at, so initial marking of the system could still (0) simply 0 right, initial marking 0 no customers in this system, so we have to defined what will state of the system is in, how see go through different marking of the system.

So, what are all the possible marking for this system, though marking for the system is the number of tokens right in this only, in this place it could be 0, 1, 2, 3 up to infinity right; that marking represents the state of the system right. So, those are all the number of items stored in this place is simply in the number of customers in the queue right or number of customers waiting in queue, because this transition implies that is there is it is always processing someone right, this transition will be fired whenever it finishes a particular, so this will be not fired this queue is empty as long as queue is non empty it will simply pick up their next process.

And now, that is it clear why this is an M/M/1 queue and implicitly of not really defined one more think, if there are say three tokens in this queue right, which of the token get to selected for processing, that is again schedule in discipline, so implicitly we are assuming it is FCFS, but we can also bring in other scheduling discipline. So, we are saying that by this definition as long as there is 1 packet in this queue, I will start processing it and then if there is more than 1 packet then this this representation does not tell as anything at all this simply says one of them will be selected, but the idea is that it is a FCFS or could be some other queue 2, that is something will have to bring when we do the analysis part of the system.

So, that is so called as sequential action where you know one action face finishes it goes to the pipe line right simply keeps processing these are all the processing steps in the pipe. So, you

can see, you can know use this representation for modeling a pipe line for finding a threw put of a pipe line, finding are where the bottle neck in this pipe line is all those things you can actually tried the find out.

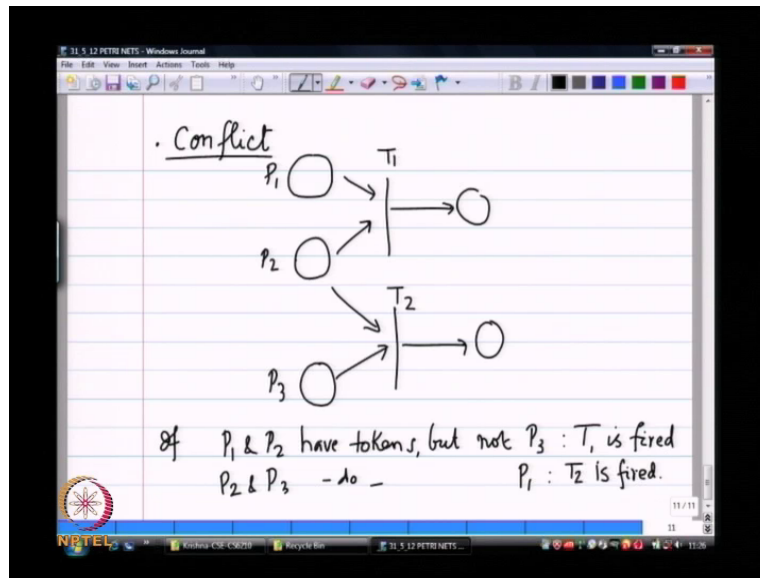
So, then we have dependency, so dependent transition is what we are seeing before, when there are more than one input places, so this is when all these places have token at least one token then you will generate, then this transition will trigger and then you will go and process that. So, this let say that this means at some data, let say now done a parallel have done some parallel processing, that is some processing here which I am not showing and let say that I am searching for something right. So, I given the search item to some different databases and they allowed to come all have to come back to me with the respective results in the corresponding databases, then this will represents the join operation right.

If you will look at process for cannot join, this is like a join operation were as long as all these places have the corresponding data values I can take them and start processing, right that was this represents. So, if you look if you look at the multi threaded operation that is what this, these are all different threads they have completed and place some data item here, for further processing and this will not proceed until all these are having at least one token then it will proceed for that right.

Now, there is there is implicitly semantics right, some of the semantics are not clearly stated from the diagram itself these are all the implicit semantics is only differences compared to see UML or may be it is not a fair compares and at least in this case this semantics are well defined, once the semantic is well define then you can implement system it will understand in bred in process given petri net and then give your results right.

Only one the semantics are not clearly defined your problem right, when we for example, right draw around diagrams, we will have different implement are a inter production for a same diagram, but here the inter prediction are fixed this is the semantic for this particular syntax it were trying here, so queues the dependency class or dependency form of transition, questions.

(Refer Slide Time: 36:44)

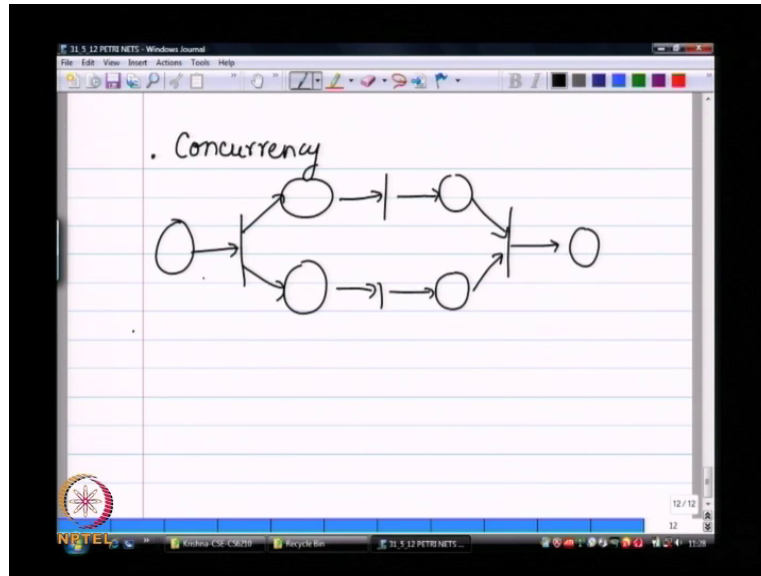


Now, sometimes we end up with conflict situations, which is also not uncommon in programming systems. It's well known what could be a conflict, so here is a transition that is expecting input from two different places and if that's right, this is one transition; now if I have another transition which is also doing this same thing, except at there is the extra line to two transitions, but one common place between those two transitions. Now, if there is only data here and here let us call it P_1 , P_2 and P_3 right, so if P_1 and P_2 this is T_1 and T_2 right, so if P_1 and P_2 have tokens, but not P_3 , if P_3 does not have a token then what will happen, T_1 is fired right, because T_1 conditions are met likewise if P_2 and P_3 have tokens, but not P_1 then what happens T_2 is fired.

So, that is clear, this system no how do resolves this what happens if all three have tokens, at given incidence of time all three have tokens, then what should happen that is the conflict right, so now both these transitions can be fired, so now there is to be an additional policy, some other specification that either says T_1 has higher priority than T_2 or some probability has to be defined saying T_1 probability is 0.05 or 0.09 T_2 probability is 0.01, so there is a probabilistic way of transition triggering a transition then there are multiple simultaneous transitions in conflict right, you cannot have multiple transitions at different places of the network then there is no conflict, when they are conflicting then you have to have a conflict resolution also right. So, that is that is again these this you will find right, you will

find in this cases when such as in occursthen there is to be some other mechanism that specifies out to resolves this particular conflict, is it clear, any questions on conflict?

(Refer Slide Time: 39:14)

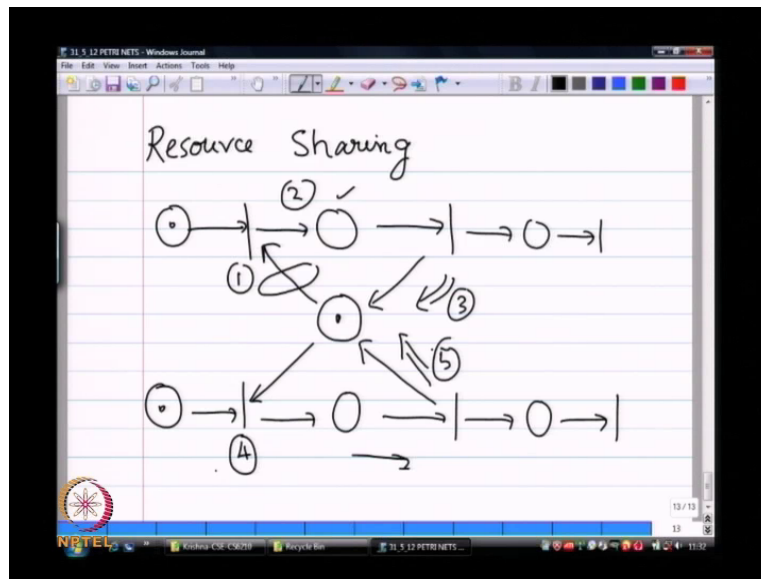


Thenow, we come to set ofright concurrency, because we say that concurrency is one of the key things is how do you model concurrency in a particular system, so let us say that have to do this parallel processing and given a value x , I have to comparewhether n or two numbers in n and x is the input to the systemand I have to compare and say whether N to the power x is greater or x to the power n is greater; I can do this sequential, I can do that in parallel right, so that was this will right to this represents. So, here now, the data is the token will contained two values tupelorange, so as long as there is one token available for processingthis transition will trigger.

And like we said before if I have something like this token will be placed in both these places right, ifthis transition triggers a copy of the data will technically go to both the places that is the semantics at a greater point. So, when any all other places will get one, one token each and what the content of the token is simply, the token it is presents some token how were that is defined right, so something that will how to defined. And so here it is, this is getting process to this is also processing right this high compute the enough x this computing x of N and then they place the results here in this corresponding places, now here is again what we saw before right, so this is like a set this is like a join operation when both these values are ready this transition will trigger and then you finally, put that in a output their; so this is how

what we call as concurrency right this (())model concurrency in the system this is this is very much all your concurrencyright operations can be model as follows, questions on that. Now, when we talk about concurrencywhat is also important some other activity show also come in to play right what is that and two systems, two processes are sharing something we have to model sharing of variables right. How do you share, how do you model resource shine, our classic producer consumer problem right, so producer is producing data, so normally what will you say producer produces data gets into a buffer right.

(Refer Slide Time: 41:41)



And then it gets arrow, now right some processing is done to that area could be something else toward the process that want to update this same variable something of that nature right. So, how do you do that, so I have two such processes that are doing some kind of processingall the transition right here, now there is a share token, share variable only thing is a need to enforce mutual exclusion, I cannot have the same variable used by these two at the same plan, so to do that we can represented this way, this is graph make any sense, so this is where the right, so what happen is right for this transition to trigger right, what are the conditions it needs in input from this place and in input from this place.

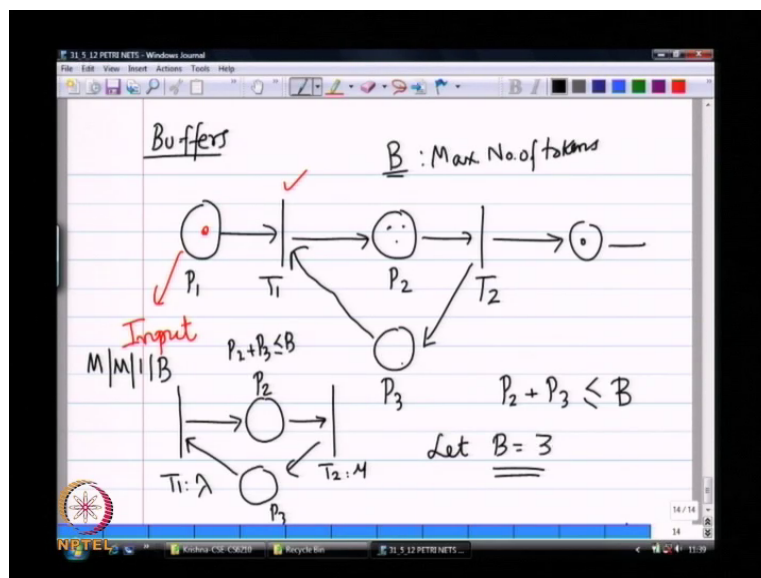
And then the result of the computation is toyed here, and after it gets toyed, then it gets I then then once there is in item here right, then it will trigger this long as one item herewill trigger this transition, now what will happen is the copy will go to further processing for this particular process, but the original data is essentially return back to the sharedbuffer right,

this represents the shared space as long as this is item here only can I this one be triggered there is only one item here, so this is the case when this condition is satisfied, now what is happening is actually conflict here right. We have a conflict here, because if this is the case which of these to going to get trigger, anyone of them in units are whatever systems, we handled at the help of new text, they both ask for the lock at the same time, one of them gets it one of them does not right.

So, assume that when this was the case that this is own that get selected, this is selected and the data gets processed it here and then it gets process here and then the copy return backs returns back to the buffer let we started, then what will happen again now then this transition can then trigger after that right. So, this is step one gets process tier and then you know comes back here, now that the data it item is available, that is also data item available here then this can get processed and then gets toyed in the next transition triggers and then after that we simply return that back, data back here right.

That is one way of capturing social, now the semantic is important right implementing this when a tool will that we have make sure that these are all clearly stated and how is the (()) going to be the done also very important; could be simply random anyone of the them which is what will happen in a mutes case right, is the one in get unless there is specific priorities associated with those. So, that is have you would handled, you can model resource sharing, so like this there is lot of other you know variations also.

(Refer Slide Time: 44:57)



So, how do you handle buffers, how do we capture a limited size buffer $M \times M \times 1 \times B$, how will be model that if you remember, count the put some restriction on the number of tokens, so has to be some other semantic its comes in, so let us try this particular things. So, here token is produced and the token normally gets toyed in this buffer, but let say this buffer there some capacity, this a maximum number of tokens available, so here is some input buffer where we can start of store endlessly. So, this is P 1, this is T 1, this is P 2, but I want to make sure that in this P 2 part of the system I should have exactly are at most B buffer I cannot handled more than that this is what happened this is like processing right arrival comes in, this is processing are the server is selecting the packet and then once the processing happens it goes on to some other part of the system right something that on here about.

So, what we do this, when P 2 finishes we introduces in other place called P 3 that will feedback to this T 1 similar, the resource sharing example we saw before, now the condition is that P 2 plus P 3 right should have less than or equal to B, this is way I can limited the buffer in the system. So, this again something this is part of the specification right, so have to normally draw this diagram also put in some other data that explains the, this is the meaning right, so have to some other notations to capture that. So, if I do that then what happens initially what where will lets a B equals you know is 3 let B equals 3, initially what will be the system state initial marking where will those three tokens B P 1 inputs in coming here. So, that is some input process it is generating this P 2 is time variable, this is where this is actual storage before it goes to server right. So, this could be infinite I am not thinking about that this is where I want to control and you want to see at most 3 packets in P 2, P 3 combined. So, were we will puts the three tokens P 3, so I put the three tokens here, I put the three tokens there right. So, this is empty now what will happen can this T 1 fired, T 1 cannot fired unless there is the inputs that in generated right, so I will try red, so input arrivesthis is input, what happens now, I will there is at least one token here, at least one token here and therefore, this transition will no fired.

So, when that transition fires, this token is removed one token is removed and where is the token now added token gets added there right, so P 2 has no one token. So, what will happen is going to get processed by this transition T 2, so after processing right when this transition T 2 finally triggers and completes what will happen this packet is now processed in this going on to next destination, at the same time you are now returning right the token here, that is word try it to do. So, therefore, in the instant of time that only process this 1 packet there,

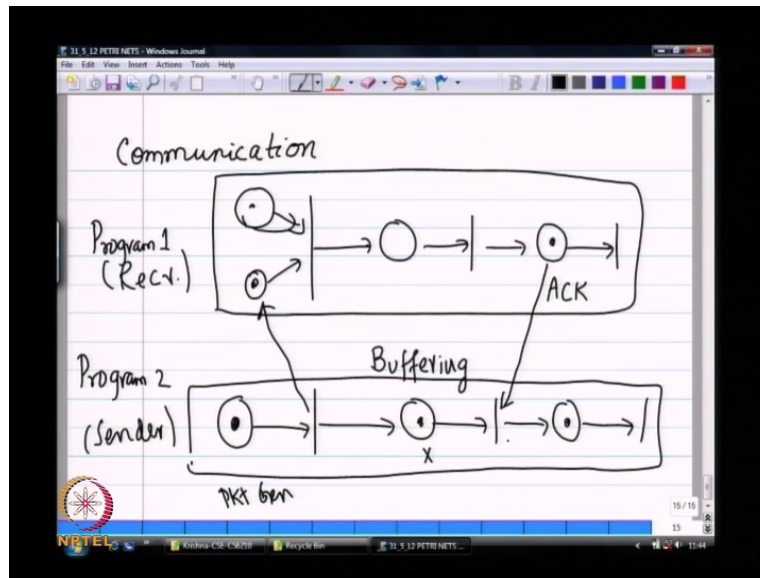
they were two tokens setting here, there is 1 here, so collectively we had 3 right, but we could also have this (()) over the opposite is also true right, so it is possible that I will have say no packets here and let say that currently.

I have 3 packets setting there, that looks like a smiley, so there are 3 packets sitting in P 2 which means there is no packet sitting in P 3. So, what happens if a new packets comes in can it trigger, new packet cannot get trigger why because, there is nothing in P 3, so when will P 3 get one token only when one process that is or one token that is getting process T 2 finishes and comes back to P 3 that is you have a model a system with buffer is it understood right. So, that is need pay of trend to, but only these have to mechanism is specifying there this P 3, P 2,P 3 or tie together and there is extra condition right, so this is now additional distains on the system that, we have to tell the underlines in implement under than underline tool to have able to process that correctly; so questions on this part, no?

Now, can I simplify this to represent by M M 1 B with what remember me saw M M 1 right had just a two transition then one place in between, now can I do same think how do I convert this two way M M 1 B there anyway we can do that, I do not need an input place, I do not need anything to whole the input place right (No Audio From 50.24 to 50.32). So, this is the transition T 1 with parameter lambda exponential rightthis is theimplicitly exponential, the actually special class of network this not a general, stochastic petri network that is when this will apply, but we will defined those things later on. So, then I have that, this is where the all packets of getting stored, this is where all the packets are getting processed, this is my secondary buffer that is all right this represents M M 1 B as long as will still called P 2 and P 3 right where the condition that P 2 plus P 3 should be less than equal to B, that is all any right, because this tried trigger will even there is no(()) keep on triggering right every one more lambda units of our main units of time it will keep generating a packet, and the server will keep processing after processing we do not care what happens, you only care about the number of packets in the system for given lambda and given mu that is for we want to look at.

So, that is your, how we can capture buffer in the system M M 1 B right, what we saw before was only P 2, but not the extra P 3 is brought in to capture the limits on the particular system.

(Refer Slide Time: 51:57)



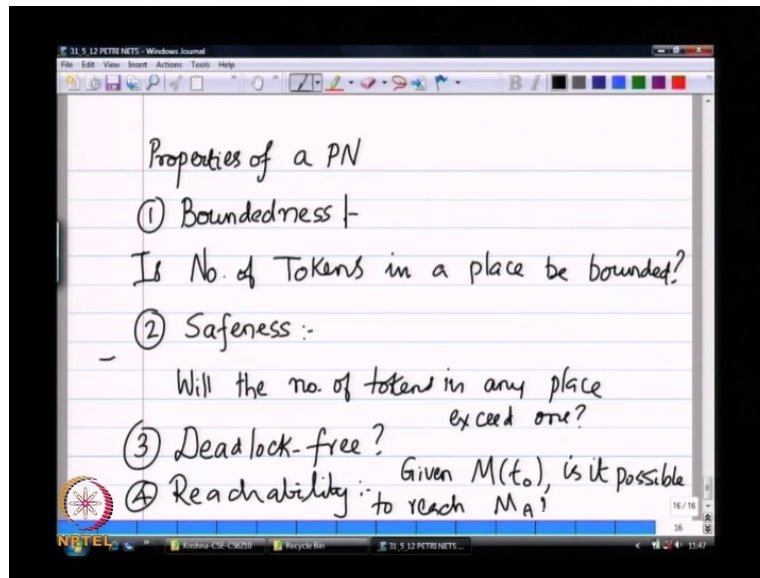
And likewise, we can also you know when you talking about communication protocols, network protocols we can also emulate or model the passage of messages between two different programs. So, let see there is program 1 or process 1 let is doing some set of processing right, so that is the conditional class there is right there is the dependency there and then some set of processing is taking place and this is one process program then there is program 1, which is doing some processing it generates something and output of this transition goes there and notice this around the put always there right side right, we can enough do wherever want to as long as the arrows clear we know what is this representative, one more transition, one more place and then finally, this is done. So, this is one message that is going from program 2 to program 1 and we care also have and other message from from here to here sorry, this is a way of trying to show two programs sending messages to each other. So, what will be the interpretation of this, can you think of let say this is the sender and this this technically need not be there this particularly loop actually lets need not be there. So, what happens is this, so when the sender whenever there is the message to be sent this transition triggers, so message is available transition triggers.

What is it do, it sends that message to other program, so basically there is a token being placed here, and let say the token is always there right it is not relevant. So, as you as get a message this is sender, this is receiver right; sender this is the processing of the sending packet all the transmission of everything is happening here, messages now low read there and

I can introduce delays on this arc itself, which here not at you know dealing with for now. So, now, there is a message available this is some other internal condition which is true, so message gets processed by the receiver and then the message gets received and then right some initial processing is done here, then some computation is also done here, after the end of the computation what happens is right, let say that there is a tokens it finally gets placed it could represent the arc packet.

So, this represents the message being sent right this is where the packet generation is taking place, after sending the packet what to I do with it I buffer it right we all buffer it right then I sit here waiting, so this transition will not trigger, will trigger only there is a buffer packet and an acknowledgement come from the receiver I then go forward and I can remove this token from here, place the token here right the buffer can get empty, that latter only when get the return message in terms of an acknowledgement, this is like a simple stock, when protocol I send the message wait for the sender to the acknowledgement, I get the acknowledgement and I go forward remove the token from here, put the token here this could be simply, post packet processing right, whatever that is involved here this could be deleting this packet recording delay and thinks like that. So, this is the way and actually model the communication between to entity in a given system right. So, these are all the various things that you can do for a system is its communication mode explanations are fine, so packet gets generated gets and gets processed some additional processing acknowledgement get generated comes back right. I could may be removes this place it is really not relevant right, just to show that some activities is going on basically I get it process technology its sent it back I can release the packet from I buffer from the sender buffer and then proceed for that beyond that point. So, like this there are lots of such combinations, so now what do we do with always, so what is the point of all the diagram drawing right. So, we are interested in some mathematical properties of this system and then also finding out if we can derive some performance matrix from this system.

(Refer Slide Time: 56:43)

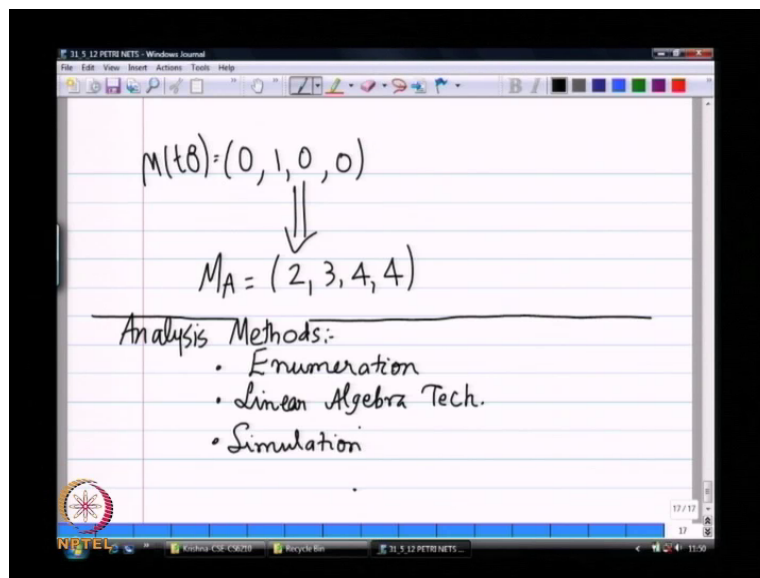


So, what are the properties of the petri net, that one would be interested in there are several we will just list about four of them, the first is called the boundedness property, we would like to know for a given system, for a given marking depending on the way all these transitions flow right would the number of tokens in a place be bounded, that is a questions that we want to ask, that is the fundamental property why because, if the number of tokens keeps growing unbounded we know the system is unstable and therefore, right being in the having lots of delays in things like that we see with the M M 1 right, MM 1 if lambda is greater than mu it is going to be unstable and therefore the queue size will grow very large and therefore, we will have right unstable behavior. So, there is something that is a property that we will like to investigate and the second is saw, so called safeness which is the special case of this boundedness property which says the system is supposed be safe, if will the number of tokens in any place exceed one, that is also an important question, will any place or a number of tokens by mistake exit one where you do not want that happen is that say this is the manufacturing plan right some assembly is going on and you do not want to have say two parts right of its tried build a car are may be some other small at component, you do not want have two parts are two numbers of same part trying up in the given assembly like on a given particular unit right, one machine can only handled say one particular item you do not want to have two items showing upon the disk at is similar can happen right. Because some have the way the delays are know done in the system is it will it be the case that there will be a one particular place with more than one token between, more than one component we will build

which it cannot handle which means system right end that machine may and crashing and so on.

So, that is the so-called safeness property of the system, which we may not relate yet in terms of computing there are probably other applications and the second thing is whether is dead lock free, in other words enter a marking where from which is in get out of we will about a dead lock right where. So, I entered in a state are a marking where I cannot leave that marking and move to any other state of the system, so dead lock is also an important property of such a system and the fourth is so-called reachability property. So, given the initial marking M not M of t_0 thus initial marking I want to know, can I reach this some other specified marking, is it possible to reach some other marking say you know M_A .

(Refer Slide Time: 1:00:11)



So, what is that mean it means this right, let say there are four places in the system and the initial state 0, 1, 0, 0 this is M of T not for some particular system that we are modeling, I want to know whether is it possible to through a series of transition at just 1 transition, it was series of transition can I end of in the state M_A which will be say 2, 3, 4, 4 are something like, that this is some for some reason I need to find out the disk and happen are not, this may be should happen or should not happen right for whatever reason I want to know whether it is possible go from this state to this particular state, so how do we do that, is it possible right.

That is something that system should be able to answer with help of its tool that be talking about right. So, these are some of the properties of this petri net that one word like to investigated, so how do we do that, I will just stop after there, so the analysis methods, so given a petri net and given this petri net on this different properties and how do we go about (()) stating the properties that we are true or untrue whatever it is we have to shows improves.

So, the simplest way to that is enumeration right I have bunch of states I have bunch of transitions are try all possible combinations right start (()) first one then T 1 then T 1, T 2 T 1, T 2, T 3 and so on. So, we can do just enumerate and there are tools that at will do that for you up towards next and right you can actually try enumerating to the extend possible become exponential then of course, you run out of time, but you like to see weather this is endurable and reasonable amount of time to be able answer this particular question, so that is one particular right there. And the second is M M sir linear algebra techniques where we use repeated you know matrix multiplications, also can be used I am not going to the detail of those due to lack of time, but these are ways you can actually used in algebra to find out right the probability going from are not probability is, that we will going to be the next state, we saw this when we did our Marko state chain Marko chains right. Where given the initial state of the system that and then gave in the state transition matrix right, if applied once that is once to transitions I apply the multiply by the same esteem again I get the next possible set of states and which this keeps on going and till you find out the steady state probability right, that is how we did that at the similarly, we can keep on try to this the algebra techniques of this just multiplying by matrix over and over again find out some set of properties of the system itself.

So, that is something it we will again not cover about and the third is actually simulation, so, this is gain of interesting right we said going to use petri nets to complement our simulation, now I am saying actually I will simulate the petri net to find out these behavior of the system, so in some cases what what the idea is that you can either this the same code right, can be coded as the petri net with this more formal representation.

If you look at this your simulation code is there of formal representation of states and transition between the states we do not really have that right, it is bunch of function is called in like that, but assume you can represent that in a formal manner with this transition in things like that, then you can actually bring any simulated right, we did this all those one of the first assignment, the tutorial the assignment one was actually simulating the change of states right,

we started of some states with some probability p_y we move to different state then to different states and so on, that is simply (()) simulation right.

So, likewise I can do that, I can start of with the initial transition with initial marking and then when there are choices right I can keep one randomly choosing one of those choices and then tried similar system behavior and then see whether it is possible to go to this particular state are are some other state in the system. So, that is also these are all different techniques that can one use for analyzing the system right and then that last techniques is also converting this petri net are the will at stochastic petri net, so we can convert as stochastic petri net to a mark out chain a continues time mark out chain then you can see a continues time mark out chain properties to actually solve problems are answer problems in the original stochastic petri net. So, that is the first level introduction of what petri nets are, so it basically collection of places transitions and arcs at the system defined behavior that is what we are seeing now so for, we will come back and the next lecture, we will cover as to we look at an example, that will show actually how we can model a system, and how we can actually derived a performance matrix from those system, we from those system with the help ofthe petri net model.