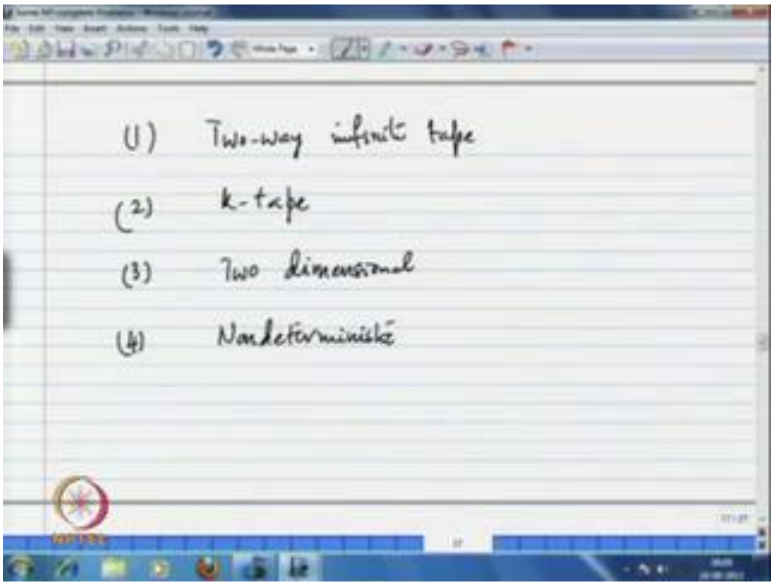
Formal Languages and Automata Theory
Prof. Dr. K. V. Krishna
Department of Mathematics
Indian Institute of Technology, Guwahati

Module - 11
Turing Machines
Lecture - 6
Varients of Turing Machines

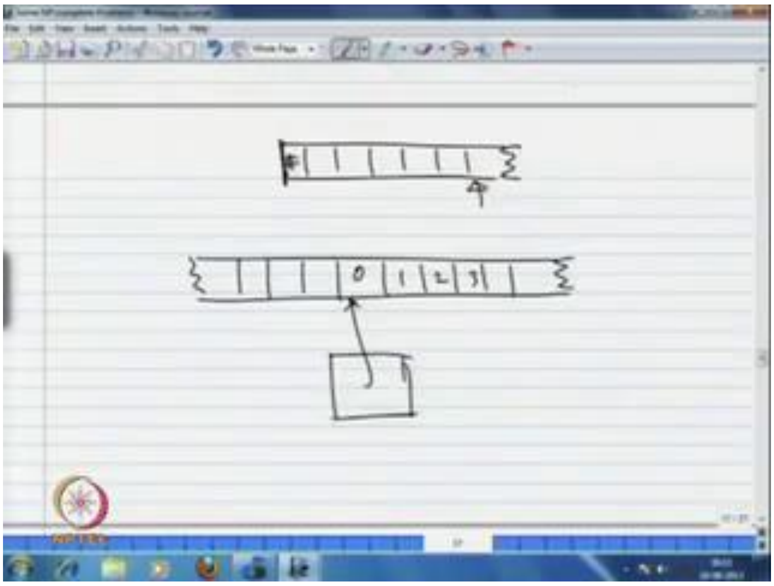
In this lecture we will look into certain variants of Turing machine, you see when we have discussed in case of deterministic finite automaton there also we have looked at certain varients and which motivates you to Turing machine particularly. You know we have talked about two way Turing machine, sorry two way finite automata and then more milli type of machines which essentially having certain features of Turing machine. So, that is how we have discussed certain varients there here the varients of Turing machine what we going to discuss is essentially some sort of extensions. Now, having certain features which you can use in various places to construct Turing machines very easily using those features.

(Refer Slide Time: 01:16)



Among which I will, now state few of them that is two way infinite tape Turing machine with k tape Turing machine and you know there are certain others like something like two dimensional etcetera. Now, something which is very important is non deterministic non deterministic, now let me explain what are all this varients of this Turing machines what kind of extensions are there.

(Refer Slide Time: 02:17)



Now, you see when we are working on a standard Turing machine we have left just fight right side infinite tape. Now, whenever we are working you know our working nature is starting from here and we keep on checking whether the symbols and depending on the type of input what we are receiving we are always keeping. So, of course as a special symbol this blank always at the left hand are depending on the situation, whatever the way that we are computing we are always cross checking carefully whether we are received to this end or not.

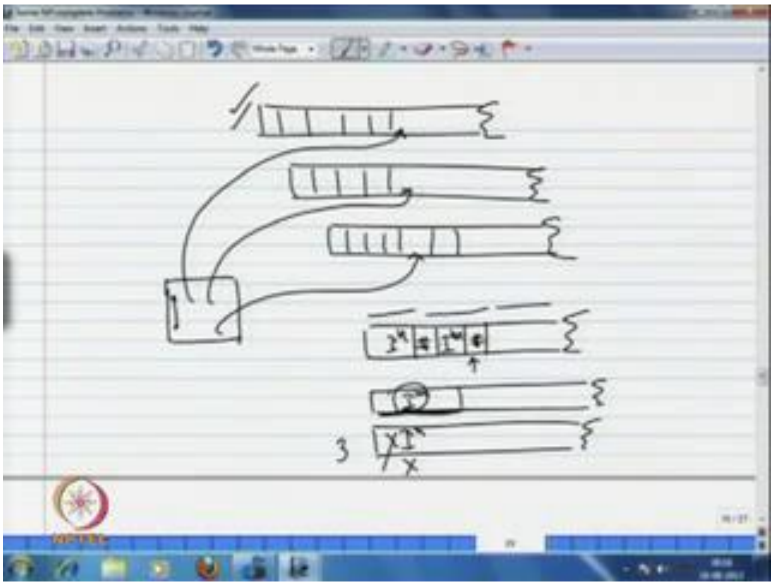
Otherwise, what will happen the machine will hang, now to avoid such situation and you know sometimes when your computation, when you are looking into each time you are cross checking. But, you are you wanted to go back and forth this way you may get better flexibility and constructing Turing machine very easily you know by taking a tape which is both sides infinite. So, that is what is the essentially the feature of two way infinite tape Turing machine because here the tape you may assume that I do not have any worry that I have to just verify.

Now, whether I am reaching to the left hand I can simply look at the computation as long as I am having, you know I wanted to cross check something without worry I will simply keep on moving to left. Now, there is no question of hanging such a feature when we are including such Turing machine you know we make all that as two way infinite tape Turing machine. But, depending on your sort of like configurations and those things when we are defining we are not worried it to look at this. So, where you are starting you may call them that as 0, 1, 2, 3 etcetera and you can put some negative minus 1, minus 2 cell number.

You can make appropriately whatever that we wanted this is what so called this two way infinite tape Turing machine and it is all working nature is almost same. Here, only worry you like, you do not have to worry about what is going to happen when during your computation whether this will hang or not that kind of care we need not to worry. You can formally define it because if you look at the standard Turing machine when we have defined m is equal to we have written Q sigma delta Q naught quadruple states set sigma input alphabet delta the transitions you know.

So, this Q naught of course sigma has special symbol blank, here also you can consider the same thing and the transitions are also the same only thing is when you are talking about the computation. Now, you can have better facilities exactly like standard Turing machine you can give, but that is all. But, when you will come to the second one, the k type Turing machine what is this how it will be, this will be like you know you will have.

(Refer Slide Time: 05:31)



So, I can like this k type separately or I can put it like a grid whatever you want let me put for the time being separately. So, say some if I considered two tapes or three tapes whatever like this am I put and here the finite control again only one control. You will have because for a one Turing machine, you have only one control it is not correspondence to each tape. But, here you will put you know will have heads like this connecting to each of this tape you will have k heads.

So, reading and writing head the finite control is essentially one only and the states you will have in this the internal states one sort of states only. But, what is the working nature we can fix some convention you know I can always take the input on the first tape. So, if I have to give some output, if it is you know for a for the purpose of Turing computable function. But, if it is computing a function you require an output I can expect always output on the same first tape and, now what is the use of the other tapes the other tapes can be used for some rough work essentially.

You know depending on the type of problem that we are considering the other tapes can be considered for that purpose and I will work on this k type Turing machine likewise. Now, you see for certain examples you can have very nice facility when you are doing some multiplication or whatever say certain amount of information if you wanted to keep. Here, what we used to do we used to go till right end and we keep using those cells and making all that these things. So, for example if you wanted to have some multiplication you do not have to worry to once again come back here and copy, come back and copy that is what we have discussed.

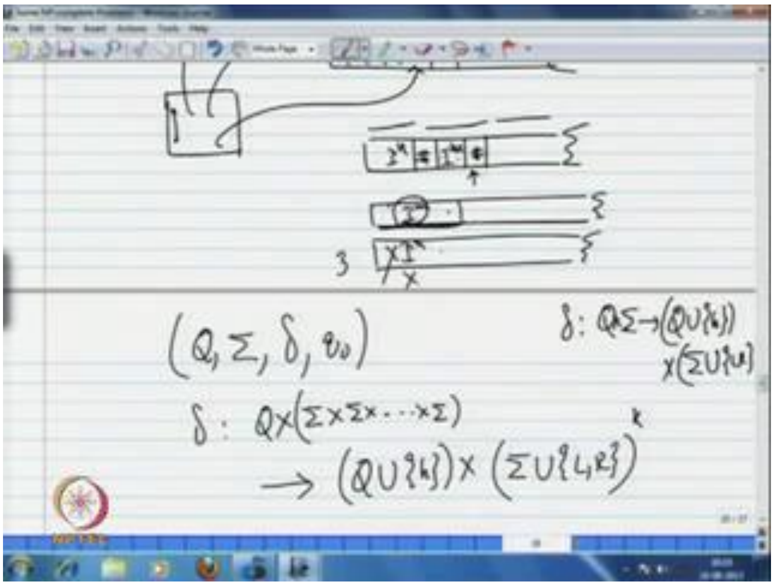
Now, what you can do see for example if you are given two numbers in unary format, let me take that on first tape say for example I power n, say blank I power m say two numbers when you are given we were starting. Here, what we can first do is this one of the numbers that we will copy into this tape right say for example I power m I have copied here.

Now, what I can simply do even I power m if you want to copy to another tape, so you copy that, now this candidate when you are reading from one side to another. You can simply type on the first tape I power m, once that you can mark simultaneously here once again when you are reading this simultaneously you can type once again I per m.

So, second I you can cross here and I power m, likewise say for example if you have number 3, here three times this I power m can be copied very quickly. So, you see the complexity will reduce because earlier what we are doing we have only one tape, you have to go back and forth to see that this number of i are to be copied. So, how many times this number of I, say for example I power m is getting copied how many times that you have to copy you have to check. So, that means several times you have to go back and forth and see like whether these many times I have copied it or not.

But, now what you do you simply you can simultaneously do because you have various heads here, similarly there are many other examples that very important one you know with respect to universal language. Now, we will be constructing universal Turing machine that is three type Turing machine that is a very good example and you see by giving such a facility you can in fact construct very nice a simpler Turing machine. So, which look very simple the simulations that you can show very easily, so such Turing machine can be constructed with a extended feature now how to define this formally.

(Refer Slide Time: 09:35)

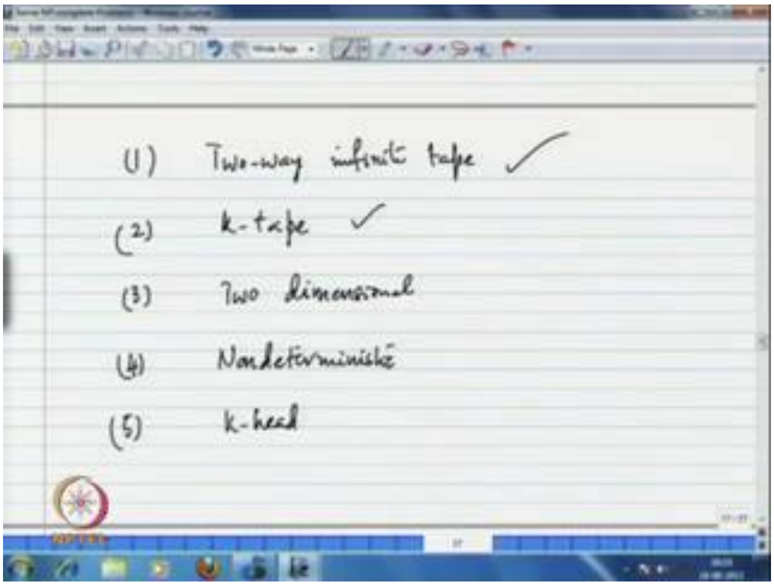


So, here again internal states you can have the same this thing you go one set and input alphabet again the same thing. Now, the transition map delta and Q naught how should be this transition map because this transition map is from states earlier what we are writing in case of standard Turing machine. So, the transition map is Q cross sigma to you know you can have halting state also and then when we are looking at either you can print a symbol or move left or right.

So, that is what in the standard Turing machine we are writing, but here you see what is the input can be from the first tape can I mean input the input to the Turing machine we are always keeping on the first tape. But, in a transition you will be reading this simultaneously you can read what is here simultaneously you read what is here that means. Here, this sigma you can take k copies of this because in all the on all the tapes what is the information that you have that you will read. Then you will chase the state right so this to q union this halting you may get halting state and then what is here in each of this tape, each of this tape.

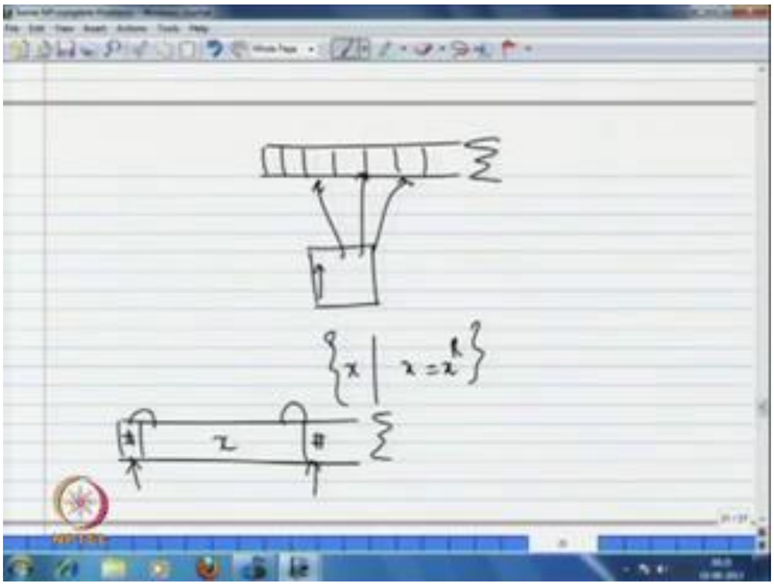
So, either you know you can put some symbol you can print some symbol and other tape is simultaneously, you can move may be right or in another tape you can move to left. So, that means whatever that we have here sigma union l r, so this candidate in each tape individually you can do whatever that you want I mean this k I am putting. So, this k times one is each of this is corresponding to one of these tapes, so we can formally define a k type Turing machine this way and there is another varient I have not listed there, let me put that.

(Refer Slide Time: 11:44)



So, you know k head Turing machine how this k head Turing machine.

(Refer Slide Time: 12:00)



So, I will be having one tape, but you can put k number of heads on these tapes like standard Turing machine I may have one tape. Here, always finite control is only one the internal states you have certain states here pointing here you can have some heads. Simultaneously, you see these are all virtual machines right this is this kind of machine you do not have in your laboratory you will you will essentially a simulator this one. Now, you do not have any problem that you know whether one head will go and the other head will get jam, this kind of situation will not occur.

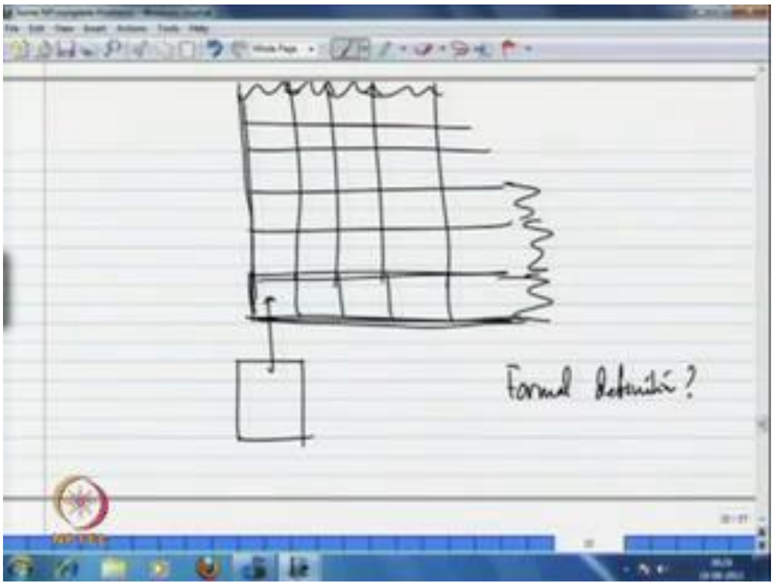
So, there is no problem of that that kind of thing, now by having these many heads what kind of flexibility or what kind of extension what kind of facility that you can have you see. So, for example if I wanted to look at the language set of all those strings x is equal to set of all those strings palindrome. Now, for example x is equal to x power r whether a string is reverse of it or not, now if you consider the input on this if given x to cross checks what you do one head you put it here other head you can first move in the first place to this.

Then you just come from this side one cell and from this side also you come to one cell and see in the internal memory you just cross check whether these two symbols are same or not. Otherwise, what we are doing you are moving till the end of the tape and reading the symbol and then you are moving till end of the other side and cross checking by storing this in the internal memory through some state. Then you are cross checking whether these two symbols are same or not you see the complexity you have to go back and forth on the entire tape twice.

So, you have to go to that end and come back to this end to cross check whether these two symbols that is what in every look that we are doing. Now, if you have two and if you have k heads, now I can choose 2 heads k equal to 2 and then simultaneously I will be able to cross check and see that this can be easily pursued. Similarly, is the multiplication also when I do not require even that many tapes if I, if I am allowed these many heads like in the previous example what I did we have copied those the input number m and n on different tapes or whatever.

Then we have generated I power m n we can generate I power m n on the first tape as output, so here what you what we do if you have. Now, for example say 3 heads, if I considered one is tracking this I power n other can track I power m and then other head can take care of the printing part. So, that way this kind of extension will be helpful in giving Turing machine very quickly, now you can try formally defining such a Turing machine k head Turing machine the way that I have mentioned for k type Turing machine. Now, you can try formally defining k head Turing machine now what is two dimensional Turing machine.

(Refer Slide Time: 15:14)



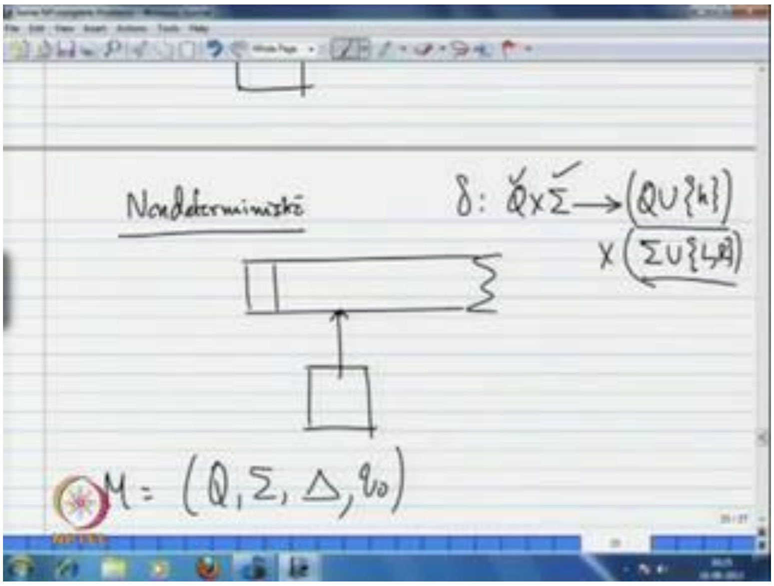
So, here the situation is like you know in your organ plane first quadrant you know a grid of tapes all the sides you know this is an infinite tape we can think like this the cells.

Now, start you will have one head to this now you will have a possibility of moving this head to up left right down. So, this kind of features will be given, now the input may fix that I will be taping taking the input on this first grid this sort of tape.

Then you will using this entire grid here all the sides you have infinite because this is left justified of course and of course this is bottom justified. Now, the grids you see the cell this is something like your first quadrant organ first quadrant of the organ plane you know we are marking the cells it will, it will look like this.

So, in this 2 dimensional Turing machine, I have to this was 2 dimensional Turing machine you can now use this cells and appropriately you can use these cells for the memory and the complication can be reduced. So, this kind of two dimensional Turing machine is another varient that you can also give a formal definition of this kind of variant, now let us come to the last varient that I have listed here non deterministic.

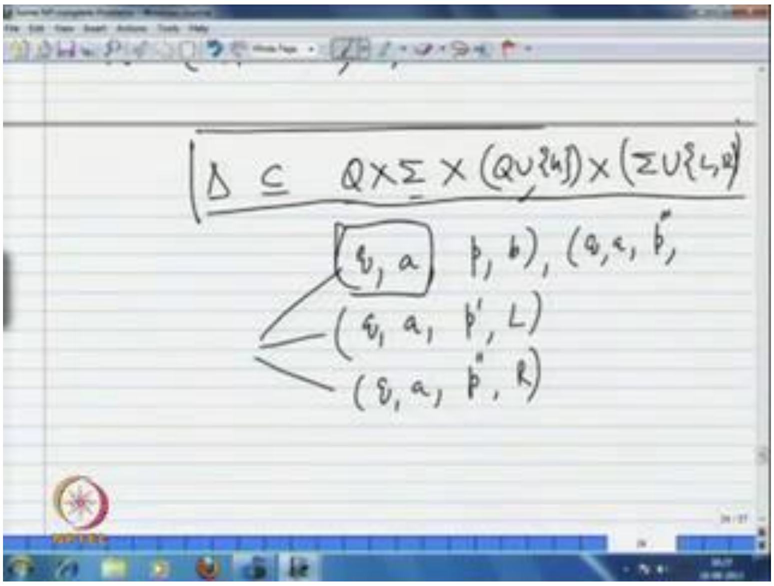
(Refer Slide Time: 17:00)



Now, you see this non determinism is already familiar with you in the context of finite automata right I have pushed down automata, so non determinism is essentially in case of deterministic Turing machine. So, once again let me write this is h delta Q cross sigma then in given a state and a symbol what we are doing we are giving a state including a halting state or print or move the head to left or right. So, that is how we are having this assignment, but here what do, what do we do this will be like in the standard tape only.

Here, we do not have any change of that but, only thing is that though simulation that computation will consider non determinism here that means if you write here m equal to Q sigma, let me put this delta the big one Q naught. Now, what is this delta, this delta given a state and a symbol in determinism what we are doing we are giving a next state and something we are doing corresponding to that state, here we give finitely many possibilities.

(Refer Slide Time: 18:29)



So, that means this delta will become a relation it is a finite subset of Q cross sigma cross you know you can give a next state one among the halting states are this cross that sigma union L r when I am writing this you see given a state and a symbol. Now, I can assign one you know this corresponding to say q a for example I can choose some p and I can say. So, for example print some b that is one possibility I can give this is a relation and then q a, I can also give say p prime and ask it to go to left and for q a, I can take some p double prime and I can ask it to go to r.

Now, I can also it is q a I can take say some p triple prime and I can print some other symbol so you can give various possibilities among this. So, that means here I am considering this as a subset of this that means a relation right, so the non determinism can be given as a relation and in that is only difference. But, of course which makes a lot of difference that means non deterministically you can choose one of the possibilities that you are going to assign among the possibilities given to a particular that q at his is what is non determinism.

Now, you see the facility will automatically increase right we have I have explained in k type Turing machine or 2 infinite tape Turing machine in non determinism. So, you know already in case of finite automata how flexibility that you have how quickly that you can give certain finite automata. So, similarly here also non determinism will help you to give Turing machine for some of the languages very nicely very quickly very easily.

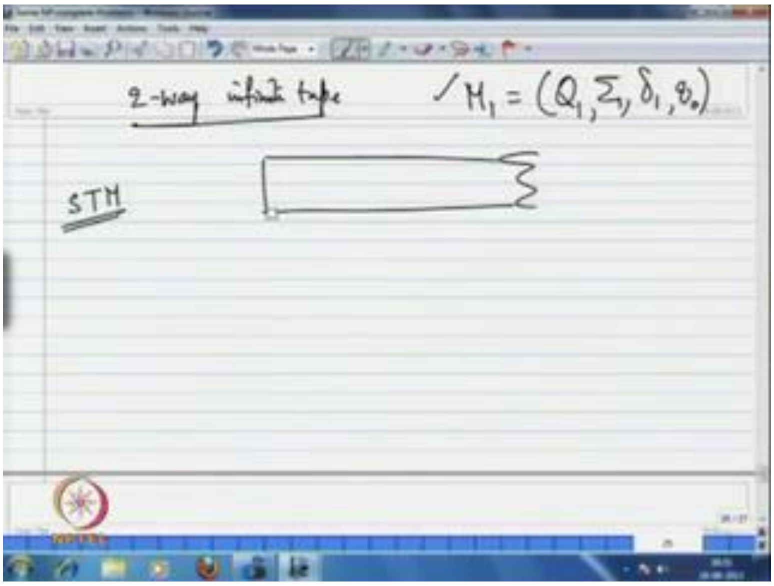
Now, what how does the computational power among this varients among this extensions would that varient if you look at that question the point the answer is there is no difference in the computational power between standard Turing machine. So, in of the varients you know in case of finite automata we have that situation in case, in case of finite automata whatever the language acceptable deterministic finite automata. But, it is same thing as languages accepted by non deterministic finite automata they are essentially regular languages.

Similarly, in case of Turing machine also of course Turing machines is not just as a language acceptor we have this as a computation device that means you can have certain functions can be computed using Turing machines. Now, the same thing here if you have a language acceptable by standard Turing machine that can be of course accepted by any of these varients. So, that can be easily covered in any of this varients with their extensions and, now the converse if you have a language accepted any of these varients can we have a standard Turing machine to accept that the answer is yes.

So, that means as where the language acceptance is concerned these are these are not something like you know, you can have some bigger class of language accepted by this automata. Similarly, computational computable functions any function which can be computed using any of these varients can be sort of like done by. Here, I mean computed by a standard Turing machine that means as where as the computation power concerned all these varients are equally you know they are.

Now, they are powerful than standard Turing machine or the standard Turing machine is equally powerful comparing to any of these varients. Now, what I will do, I will just give way how to, how do we look at for at least one or two cases that how we can really simulate the standard Turing machine with this kind of behavior extended behavior.

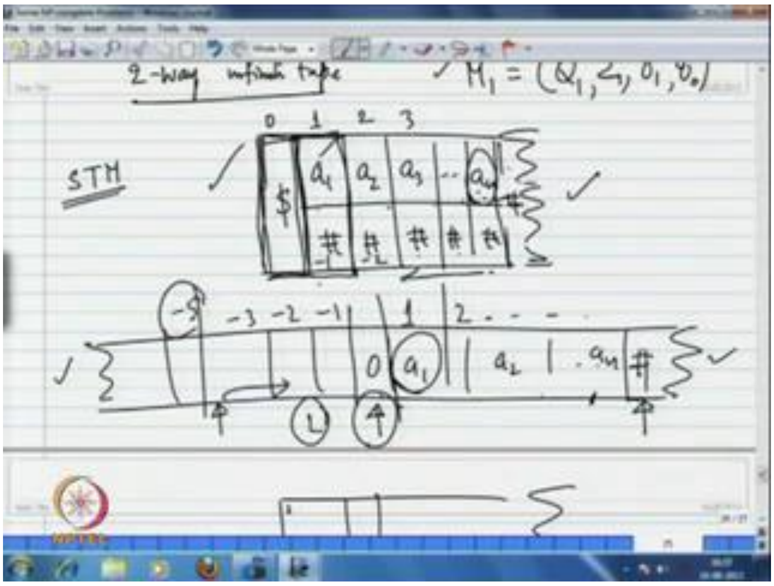
(Refer Slide Time: 22:50)



So, let me start with may be the 2 infinite tape, 2 infinite tape Turing machine simulating over a standard Turing machine. So, assume you are given a two infinite tape Turing machine that means let me call that as say m 1 is say q 1 sigma. Now, you can call sigma 1 delta 1, say the initial state, here not does not matter, now this machine we can whatever this is doing that is 2 infinite tapes Turing machine is doing.

So, we will do it via a standard Turing machine let me write S T M for standard Turing machine, how do we do that this tape I will write it formally little late. But, let me because there is nothing like you know tape width is fixed because this is the machines is not something physical right. So, what we do is, what is the input given to you on two way infinite tape Turing machine, let me parallel write this also.

(Refer Slide Time: 24:17)



So, the given one is this you have cells, so the input once it is given to you like this a 1, a 2 and so on say a n this is how probably we are working and going back and forth and doing computation on this. Here, what we do that a 1, a 2, a n whatever is there first I will change the input in the format that I can see two tracks on the standard tape except the first cell. So, in the first cell I will print a dollar symbol and then I will make this tapes I will see this tape making means I will see this tape into two tracks.

So, that my input will be seen this a 1, a 2, a 3 and so on a n and all others are just blanks, now you see what is the meaning of this because whatever that tape contain that I have I am now dividing it into the two tracks. So, that means what I will do, when I am to do this the tape is just one tape only I will have one cell only here, but this cell content I will now fill with a pair that a 1 comma blank. So, this kind of pair I will type it here whenever the reading head is moving it will, it will read the entire cell content it is not that.

So, you know reading head when it is moving it will read only this portion and it will not it will not be able to do it. So, because we have the standard tape whenever the reading head is moving you know the reading head will read the entire this cell content. So, essentially what we are going to do is what are the symbol that you have and the 2 tape infinite Turing machine. Now, we are changing this cell information that to a 1 blank of course what I have suggested is in the standard tape when the input is given a 1, a 2, a n.

So, you to make this kind of thing what I have to do I have to first shift the entire input to one cell right or you know when you are considering with the blank.

Here, you will put a dollar, here that is all and, therefore a 1, a 2, a n those are there only right and this a one we are now changing it to the symbol a 1 blank. Now, this is you treated as a new symbol you may call it as b one what is b one symbol that is a one blank a pair. Similarly, what are that a 2 that cell content that you have that I will print it as a 2 blank, likewise you know what are the input we have that input.

Now, it is transformed to pair, so that I can see it is a two track, now once I have converted this to this kind of situation what are the simulations that this guy is doing. Here, on 2 way infinite tape Turing machine we will simulate that this here parallely, how do we do that when we are taking a left move on this when you are coming to a n.

So, I will also come to the left take the left move of that head, but what I have to look at here you know I will concentrate on this input. But, I will be reading the entire thing I will be concentrating on this if I if it is keep on coming to this I can always take the left move the left move on the first track. Now, to read the first component, because I will be sort of like looking into or changing whatever that we are doing on the first tape if I take a left move, I will take a left move from here. So, let me call this is cell 0, this is cell 1, cell 2, likewise and then I will call this cell minus 1 cell, minus 2 cell, minus 3 likewise, so what do I have here one let me call this is 0, 2, 3 and so on.

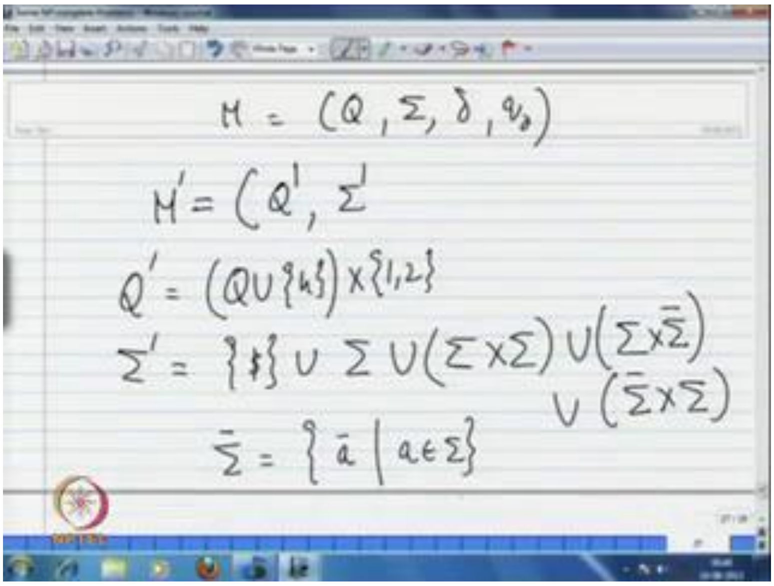
But, this will be minus 1 minus 2, likewise what I will do as long as I am in these positive number cells on the two way infinite tape if I am going left. Here, I will also go on this standard Turing machine left, here if I am going right on the 2 way infinite Turing machine I will also go on the right on this. Now, if I reach to this 0, thing I will come to this dollar, now from dollar I have to use the option of changing track that means when I am in the two way infinite tape machines.

So, if the reading had comes to this 0 cell from this cell if it is going to right that means on the positive number cell then I have to go on the upper track if it is going left then I will now consider the lower track that means. Now, I will concentrate on reading on the cells which are in the lower track or changing the cells here if it is a printing tape if it is going. So, for example from here it is going to left if it is l then what I have to do on this tape I have to go right it is, there is no left because it will get hang.

So, I have to go right and on the cell, if it is going to right say for example when you reach to this position if it is going to right then what I have to do because if you are going to right. Then I have to go to left because when I am reading coming to say for example minus 5, the cell minus 5 that means somewhere I am here minus 3, minus 4, minus 5.

So, when it is going to right on this lower tape I have to read the symbols which are in the left side of this therefore I have to come to left this kind of simulation that we will have on the standard tape. So, that whatever this exactly the 2 way infinite tape machine is doing exactly the same thing can be simulated on this standard machine, now let me give you a formal way of representing this.

(Refer Slide Time: 30:02)

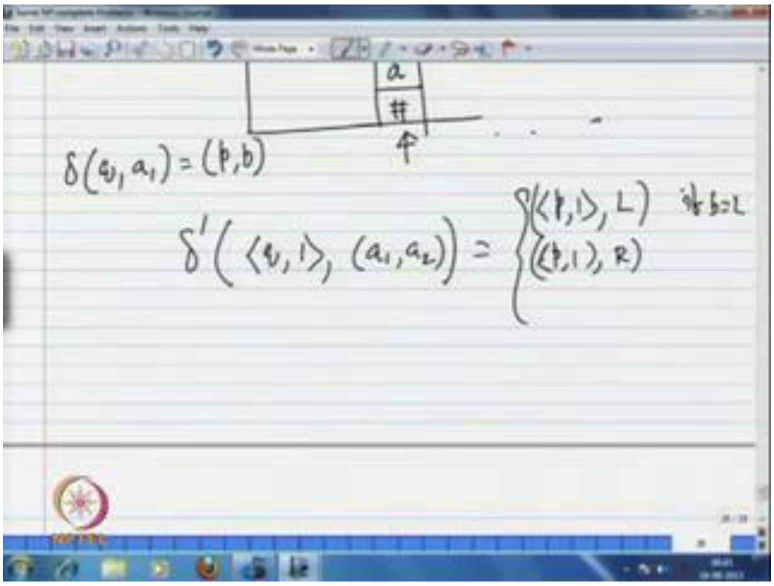


So, let M 1 equal to say Q 1 sigma say delta if you call it Q it does not matter because I am writing M 1 and Q naught it is 2 way infinite machine 2 way infinite tape machine that you have. So, I may write m prime here I requires states the states, so let me write it as Q prime what should be this because here the states essentially look for the states have to consider all the things. So, you know the state should be able to distinguish what is there on the left and what is there on the on the upper track what is there on the lower track right so states.

Now, we will divide with this Q prime Q cross 1, 2 by this what I am going to get in this Q union sigma, sorry halting state also because in the upper track if I am getting halting state that will also should also be covered. So, this state cross 1, 2 it sends that we are going to have whatever the simulations here of course the halting state will come. Otherwise, in this and sigma, now you see what do I require in case of sigma we require dollar a new symbol which is not there earlier union the original symbols of sigma that you require union. So, the symbols which you should be able to accommodate on the both the tracks that means sigma cross sigma this also you would require.

Then I take some special symbols which will record the halting that means, let me call it as sigma bar if it is halting on the lower track I will record through this union sigma bar cross sigma I will tell you what is the sigma bar. Now, this sigma bar I mean you will just consider what are the symbols of sigma with a different representation that is all for each a belongs to sigma how do I use this.

(Refer Slide Time: 32:43)



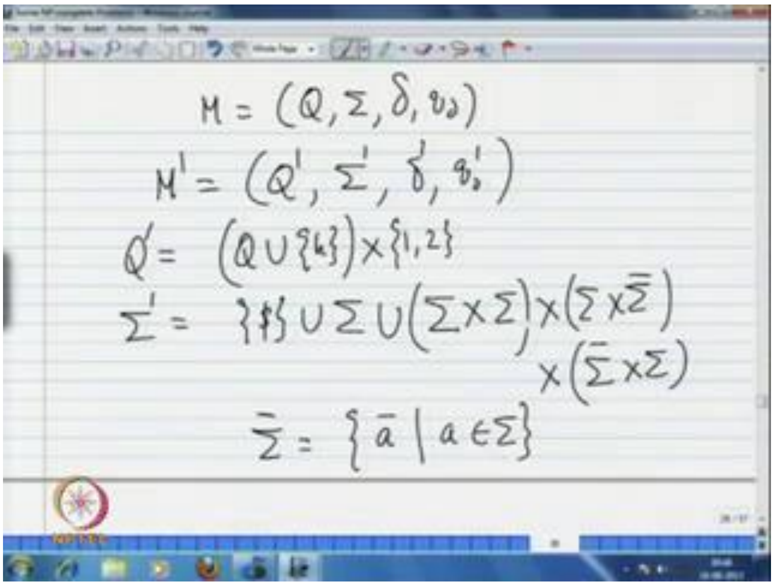
See, for example you know when I am having say for example a here blank and if the machine is halting at this kind of situation that means on the first tape if I am having a, what I will. Now, look at is this kind of situation because I have x 1, this is the current when it is halting that we will make it h 1. So, that we will understand that this is on the first tape it is getting halted and then this symbol and the tape symbol also we have to look into because this is not a halting state.

Now, because h 1 we will not consider state because for each Turing machine h is halting state, so when I have h 1 this tape content. You know where finally it is halting when I convert it back to a, I should be able to distinguish, so for that matter what we writing this is a bar will be the symbol that we will be considering. So, just to distinguish wherever exactly it is halting you know we make these special symbols, so this is what is now the alphabet that we consider. Now, let me write delta prime that I will explain you and then the initial state that we consider as it is.

So, where you are starting you are starting in the first tape, so you can declare that to be the initial state, so you will be starting. Now, the simulation part the simulation part is now first what we have to do you have to change the tape into this tracks that printing you should be able to have. Now, this delta prime how it works if you have if you have this q one that is on the first tape whatever the input that you have a 1, a 2 that is you are on the first tape we will change it to the state p 1 left if you know.

So, this I am writing in connection to in case of 2 way tape machine whatever is there that means if in the original this thing assume delta of Q, a 1 is equal to say p, b. Then if b equal to l then we take l and this is p 2 will take in case of this is in the first tape, so if it is going to if b is r we take going to right. Now, with this phenomenon I will explain how like how formally we can design a standard Turing machine corresponding to given 2 way infinite tape Turing machine.

(Refer Slide Time: 35:50)

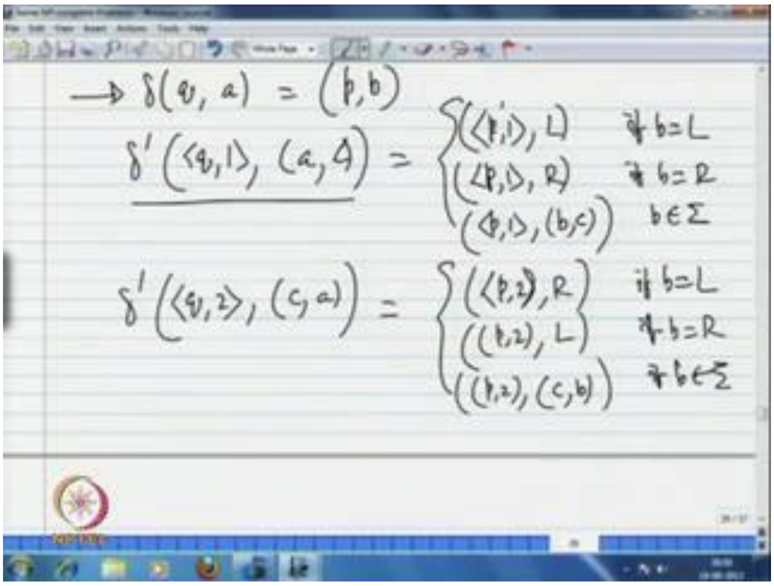


So, if you are given two way infinite tape Turing machine Q sigma delta q naught we will consider you know this M prime some Q prime sigma prime delta prime and this state. Here, we consider corresponding to you know the states of this which are which can be manipulated on the on both the tapes on both the tracks. Now, this sigma prime is essentially looking at this special symbol and the elements of this, so to accommodate the symbols as pairs like both upper track and lower track.

Then some more symbols that we include that I will explain you to look at this sigma bar I am writing. So, a tape on the first the first component is on the first track second component is on the second track the sigma bar cross sigma. Now, what is this sigma bar the sigma bar is essentially taking the symbols of sigma we just denote a bar the purpose of this is whenever the machine is halting.

Now, I have the state on when we I am on the first track I will have h 1 and when I am on the second track I will have h comma two this kind of state I will halt. But, the halt in state of a Turing machine is essentially h just when we are we wanted to halt at the same place I will record the symbol and then go to that particular place and make the halt. So, to make it we will have this kind of special symbols sigma bar which is 1, 1 symbol each corresponding to each symbol of sigma, now how do we simulated the simulation portion is this.

(Refer Slide Time: 38:16)



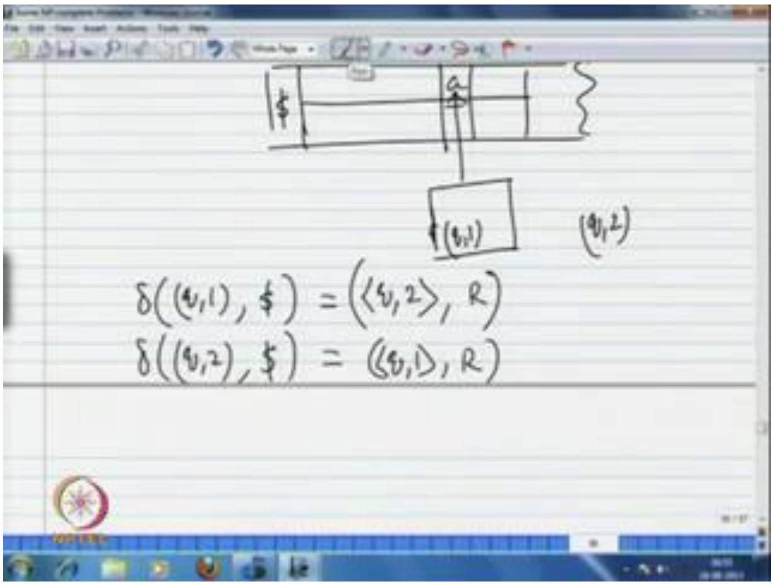
You take any transition in two way infinite machine that say q comma for example a, if it is p b, now depending on b and where you are depending on that we will be doing this simulation. So, that means this delta prime if this state when we are considering on the first track what do we do we will now consider corresponding to this a and any symbol a. So, say for example a prime should I write prime or it may be any symbol say some c on the lower track what do we do we will give that the p 1.

But, the state p 1 moving to left if this b is left, if b is right of course the changing state is this on the first tape will ask it to go to right this is the pair we will assign. Now, for example, if b is an element of sigma that means if it is a printing step the state which is getting changed is p 1. But, now on the first track only this change of symbol will take place that means b on the first track on the second track there will be no change the same c will be there.

Now, so corresponding to the transition in the 2 way infinite tape machine in the standard machine we will assign we will give the transitions this way. So, if this is in the lower track if this information is in the lower track that means the current state q and if the symbol is in the lower track that means this symbol is on the cell which is left to that.

Then what I will do, I will now give transitions for the lower track that means the state will be this q 2 and for any symbol on the upper track that is c and a we will give like this. So, if b is l as I would mention we should move in this track to right if b is right then in the lower track that means the state component is 2, we have to move to left and if it is an element of sigma if it is an element of sigma. Now, what I have to do in this though first component that is an upper track there is no change, but here we are going to print b, here is this definition clear.

(Refer Slide Time: 41:13)



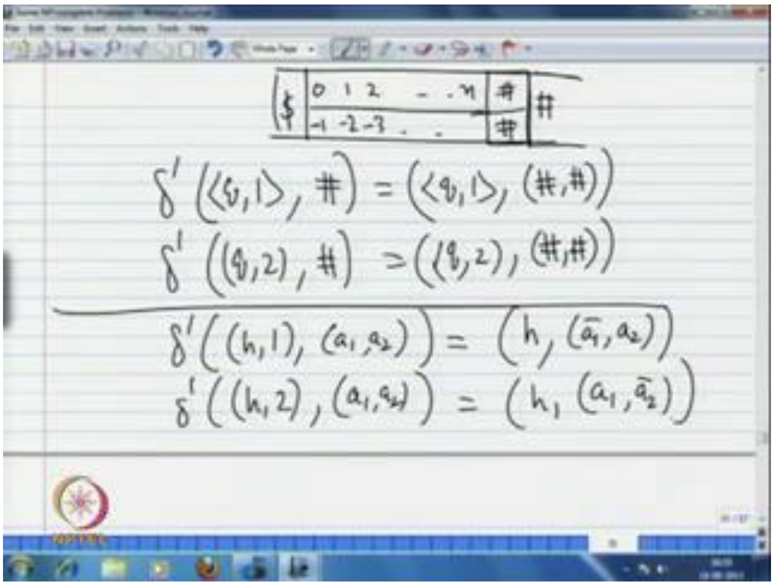
So, once again if I have to explain let me put 0, here 1, 2 the cells like this and on this after we have converted this to you know two track when I am reading a. Here, on the first track I will always have internal state corresponding to this q, here suppose I am reading the second cell. So, in the state q this first the tape the cell will be on this entire cell of course the reading held will be, but in the internal state through internal state we will recognize that if I have the internal state q 1. So, I mean that I am reading the first track I feel that I consider that I am reading the first track if the internal state I have in the second component.

Now, say for example if it is q 2, I consider the input that on the lower track of upper track and the lower track that is what is the convention. Now, depending on what is happening in this two way infinite definition we will be updating this tape. So, as I had explained you earlier, now if a is through symbol which is on the right side of the tape that means the internal state if it is q 1.

So, the first component will get affected if it is printing step or moment will be considered on the first track with respect to the track and the second track information will not change in the printing step. Now, what else we have to do when I am coming to this dollar symbol, I have to look it that you see I have to change the track. So, that means when I am coming to the dollar from the upper track then I will change to the lower track and so on.

So, that means how I consider that delta of this q 1, if I get dollar on the tape then what I will say this you change to the second track. So, that means the internal state will be represented by this and asked to go to right similarly if delta if I am in q 2 if I am receiving dollar this thing that means I am from the second tape lower track I have come to this dollar then you asked to go to right. So, that means this q 1 r you asked because whenever I am in dollar the track change will be considered this way and then.

(Refer Slide Time: 43:58)



Now, you look at the situation will be like this you have made till this point the tracks, so this 0, 1, 2 and so on n till this point that you have considered minus 1, minus 2, minus 3 and so on, you have made it this way. But, there after you have cells like this only that means whenever you come to a blank cell what to do this blank cell first. Now, we make it into two blanks that mean if it is on the upper track that means in any state q, in the upper track that means this is internal state. So, if I am receiving a blank symbol this delta prime what we have to do in the first we are in the first track the same state.

But, first we will convert this input to two blanks that means upper track blank and lower track blank, similarly this if you are in the lower track also when I am reading the blank symbol. Then you will continue to the same state when and we will continue to the do the computation in the lower track but, you will first convert it into 2 blanks. So, that the tape can be extended to two tracks say for example if I am, here first I will not just see this way I will now consider this like this.

So, this is 1 and, now this head recording that position that the halting position recording related to that what do we do if I am coming to the halting state when I am simulating on the upper track. So, I have something a 1, a 2 what I will do, I will now convert it into this halt state, but this a 1 bar a 2 and this similarly this when I am halting that lower track. Now, you see this for example, some input a 1, a 2 that we will now this is the head recording position this halting state that a 1 a 2 bar, so to indicate that where exactly this is getting halt.

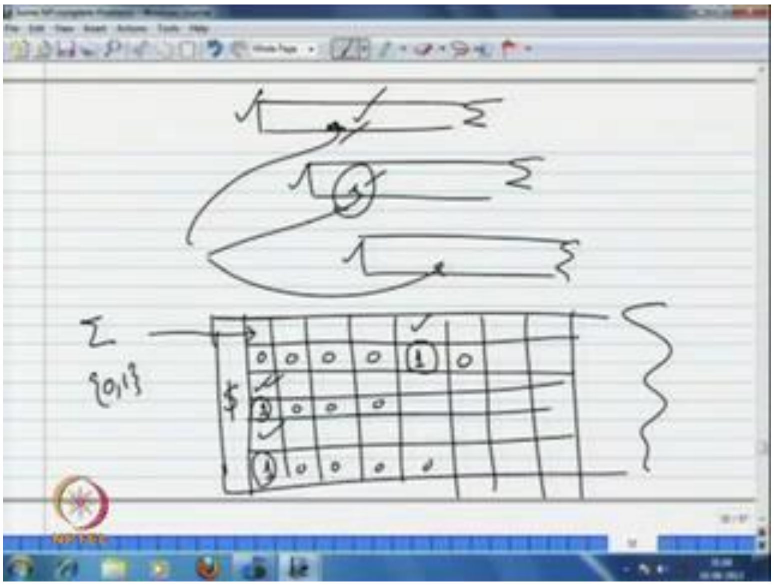
Now, you see once it is coming to a halting position I do not have to, now we do not have any possibility of continuing right only thing is it is just recording the position. So, that you know if you wanted to record this position where exactly this is halting, so that will be shown up here with this symbol. So, whether it is in the upper track or in the lower track for that matter only we are distinguishing these symbols, so with this kind of construction with this kind of simulation what we can do.

Now, what are the two way infinite definitions you are doing the job that can be simulated on a standard machine as well. Similarly, by considering this kind of tracking business that means what are the standard tape that you have we have now converted the input by as pairs. Now, if I make this as triple h you know that means say for example for the symbol a in one cell if you have.

So, I can put any symbol any complex symbol I can put in one particular cell, say for example a 1, a 2, a 3 I am putting it vertically then I will see that I am divided it into three tracks. So, this is not something like you know tape physically is getting changed only the input on that we are we are changing. So, that we can track the track the simulation whatever that we want to what are kind of extensions that we wanted to have.

Now, the similar mechanisms can be adopted to simulate when other higher machines higher versions what are the extended versions say for example if you are looking for k type Turing machine.

(Refer Slide Time: 47:55)



So, I have k tapes like this and I have k heads also right I have various heads what do we do in standard tape. Now, you divide this tape into first cell this is one cell only for just to write to accommodate this many lines I am just writing it bigger. So, I will put this special symbol like this the input is always will now make it into first track and odd number considered will be seen as the tapes that you have. Here, say for example I have three tapes here then I will divide this into six tracks then the 1 and 2 for the first tape, 3 and 4 for the second, 5 and 6 for the third.

So, what I will do, here why I am making this because I have to recover the head position also the first the odd number track will contain the elements of sigma this third one fifth one. Now, what is in between one the even number one there I will print either 0 or 1 from here if I put if I have zeros. Here, that means the reading head is not present here wherever the reading head is present there I will put one that means when I am reading.

So, I will convert first the input given to me on to the upper track I mean the first track and the remaining cells to start with they may be blanks. But, the reading head wherever it is there I will have 1 and other cells we have 0, so that is how we will first change what is called standard tape and then we start simulating. Now, how do I simulate say for example if I am doing some movement with the first head then I have to consider that. So, where is the head will be denoted by this symbol say for example 1, now first I will pursue this, here I can do it simultaneously first step. So, I may simultaneously consider with this second I can consider this simultaneously third, I can consider simultaneously. But, here what I will do first I will pursue one because I have only one reading head on this step and then see like I will transform this change. Then I will pursue whatever has been pursued by the second head so by considering where exactly, now because there is only one head.

Now, when I am working on the first step I will now first look at the where is the head position, so that I will know where is, what is the input that I have to pursue when I am concentrating on the second head to pursue the matter. Then I will now first look fourth track where 1 and then the particular input is will be pursued and when I am concentrating on the third one then I will look at where is the current head. Then pursue the concerning input, so that means what is happening here whatever that simultaneously there I am doing. Here, I will do it sequentially because I have only one head, but the when I am moving from one track to another.

So, that means if I am working with the first tape and then with second tape then I have to know where exactly the reading and writing head it was, so that is essentially denoted by this 0, 1. So, once I know that the reading head was left here you just go to that position and do the appropriate changes go back and forth. So, you may have to do so many steps for example for one step here you may have to consider so many steps here because you have to completely you may have to come from a particular position of the head to the position to search.

Now, where exactly was the head for a corresponding track to consider this simulation, so this kind of extension can also be simulated the k type Turing machine can also be simulated via standard tape Turing machine. But, see that there is no difference in the computational power of course the complexity may vary, because whatever that you can do it easily in case of this higher varients for standard machine.

Now, you may have to put little more time you may have to give more steps to do the job right, so the complexity will may vary. But, as far as the computational power concerned they are all equivalent that is about you know some of the varients of this Turing machine and recording non determinism will be studying little more in other lectures.