

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

**Module - 14**  
**Introduction to Complexity theory**  
**Lecture - 4**  
**NP-Complete Problems**

In the last lecture, we have discussed you know establishing, how to establish a problem is the NP complete. In that connection, I have mentioned two problems, which is in the lineup of the undecidable problems concerning you know turing machines. For example, halting problem that is undecidable. So, given a turing machine of course, a standard deterministic turing machine and input string  $w$ , whether  $m$  halts on  $w$ . That was the halting problem we have observed that it is undecidable. And then there are several variants that we had discussed.

In the similar lines, we have considered non deterministic quantitative analogous problem, that we have written the problem  $n$  naught. So, and observed that, it is NP completed. And then a variant  $N 1$  was mentioned; that means given a non deterministic turing machine, when you start that machine on a empty tape, whether that will halt, I mean that accepts, that halts in a given number of steps, so that we have mentioned  $N 1$ . And I have also given certain hints to observe, I mean in fact, I have shown the function that you can, you could have observed, that it is a polynomial time reduction,  $N$  naught to  $N 1$ , to understand that,  $N 1$  is NP completed.

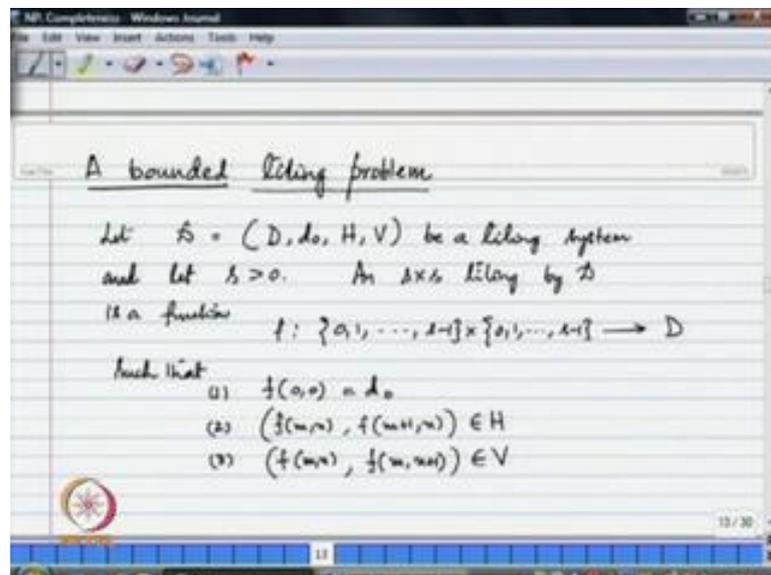
Now, this is a very nice class room exercise, problems concerning turing machines, but the power of this essentially, we can in this sequence, in this line up you know, I will introduce few more problems, I have which to establish that they are NP complete. And in fact, we can target to observe prove Cook's theorem in the class room using you know these kind of basic techniques. So, certain variances of this  $N$  naught  $N 1$  and these things we will can you can always take as classroom exercises and observe that, they are NP complete.

And now what I will do? I will start with another problem called bounded tiling problem. Because we have observed the tiling problem is undecidable, what we consider now a variant of that, that is essentially you know, a finite space need to be fill by tiles given a

tiling system, that is how will take, so called bounded tiling problem. And, that we observe it is and we observe that problem is N P complete. So, and essentially, what I will do here.

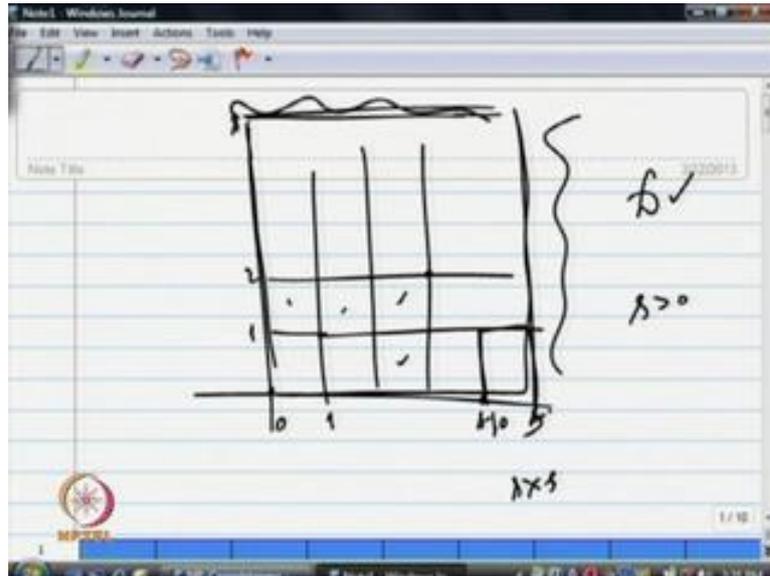
Now, what are the sequences of problems that we are establishing, because to observe a problem is N P complete, what I said, already established N P complete problem, if you can reduce that problem to the given problem, then you can observe that it is N P halt. And, Of course, we have to observe the given problem is in N P. So, these two conditions that is, how we have to, we are pursuing actually.

(Refer Slide Time: 03:19)



So, let me first state the problem called bounded tiling problem. As in the tiling problem here, you have a tiling system; that means, say this that quadruple  $D, D$  naught,  $H, V$ ; you know all these components  $D$ , finite set of tiles;  $D$  naught is the initial tile that you will be fixing it in left bottom corner of the space that you are filling. And you have certain horizontal constraints  $H$  and vertical constraints  $V$ . Now, what in addition here is, you will be given a number  $s$ , a positive number.

(Refer Slide Time: 03:51)



And the what you, what you are doing in case of bounded tiling problem, if you take the first quadrant, you are, say let me put this way, you will be say the, if the quadrant is like this, you will be choosing tiles on each position of this first quadrant, which is unbounded. What now we do, we restrict our self to you know, a finite portion. Say, this is 0 1 and so on till, so 1, 2 and so on, say  $s$  minus  $s$ . Suppose, if you look at till  $s$ , if you look at. Then, you see the tile, this portion is 0 0 0 1 2 and so on till  $s$ . Now, here this is an  $s$  by  $s$  square. So, our target is to fill this given portion.

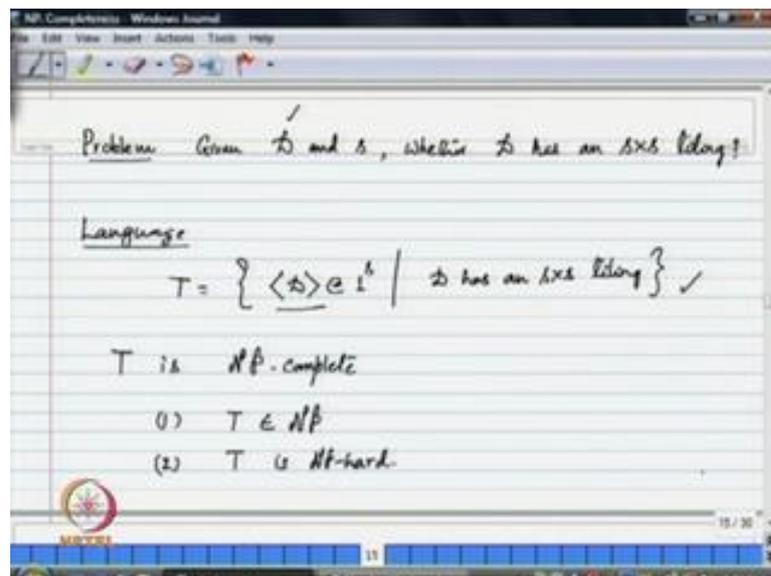
Now, what is the question concerning that, I give a tiling system  $D$  and an positive number  $s$  and ask you, whether there is an  $s$  by  $s$  tiling with with of the for the system  $D$ ; that means, Of course, you have to fix tiles from this set in this, you know the portion size  $s$  by  $s$ , that is what is essentially the target. Let me formally mention, an  $s$  by  $s$  tiling by  $D$ , this is a function  $f$  from  $0$  1 and so on  $s$  minus  $1$ , because if you want to fill this corner, I will be talking about this point only, that is  $s$  minus  $1$ , correct. So, what do you, I have here, this is a function  $0$  1 to  $s$  minus  $1$  cross  $0$  1 to  $s$  minus  $1$ , because each tile is represented by its, this corner.

So, this is  $0$  0 and this one is you know,  $0$   $s$  minus,  $s$  minus  $1$   $0$  position. So, the tile is represented by this corner, you know in tile undecidability we have discussed, the tiling. So, this is a function  $f$  from these square to  $D$ ; that means, in each position in this square, you have to assign a tile  $D$ , which satisfies these constraints of the system given to you,

what is the constraint, what are the constraints. One that, this  $D$  ought should be in the left bottom corner and if you are fixing two tiles, I mean  $f(m, n)$  is a tile and  $f(m+1, n)$  is a tile, which they are coming side by side horizontally, because you look at the position. This is,  $m, n$  position; this is  $m+1, n$  position.

So, if they are coming certain, so these things should be satisfying horizontal constraint. Similarly, if you want to fix a tile, above a tile, then they have to satisfy these vertical constraints here. So, that is, what is essentially, the bounded tiling problem, is just a variant of the undecidability undecidable problem, tiling problem. What we observe here, this bounded tiling problem is NP complete. So, to observe this is NP complete, Of course, now you can ask me like, how do you represent this problem?

(Refer Slide Time: 06:53)

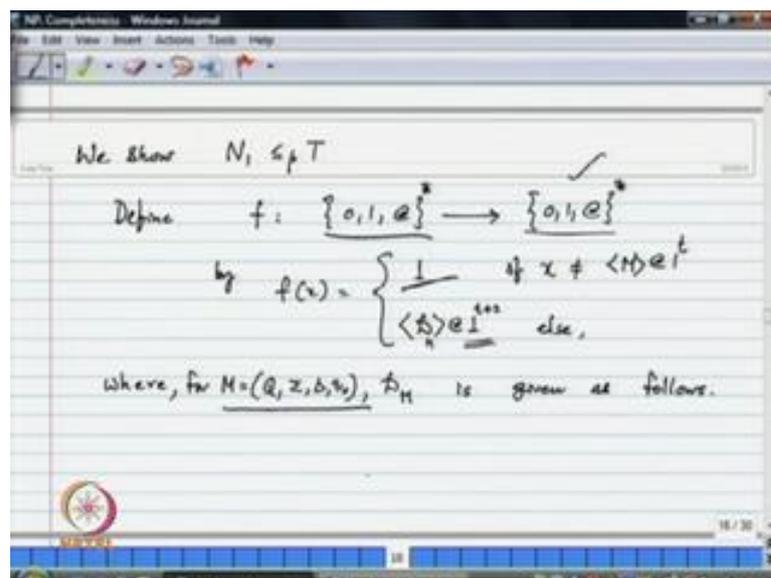


The given a system tiling system  $D$  and a number  $s$ , whether  $D$  has an  $s$  by  $s$  tiling, that is our problem. And, one may write an appropriate coding for this, because this is a finite system. So, bracket  $D$  I am writing, whether you will be coding with 0 ones or whatever. So, I call this language  $T$ , tiling bounding tiling problem. So, the encoded system, the string bracket  $D$  at the rate on  $s$ , so that time, the number of steps that you are allow to, you know perform, is say for example,  $s$  number of... Here,  $s$  by  $s$  tiling, whether it is possible, that is what is the  $s$  is the parameter of the tiling size. So, the question is right. So, you collect all those strings, if  $D$  has an  $s$  by  $s$  tiling.

So, let us consider this language. Now, to observe that, this T is NP complete of course, you have to observe that T is in NP. Now, what you will do, you have to give an appropriate polynomially balanced language t dash. That means, given a certificate; that means, if you are given a tiling, you have to cross check, whether that is an s by s tiling or not, because, s is a parameter given to you in the input. Now, you have to cross check all the tiles, whether they are satisfying the given constraints whatever or not.

And essentially, as we were doing, what we have to do, you have to write the appropriate an language polynomially balanced language t dash and observe the T is in NP. We can take this is, this portion as an exercise. Now, what I will demonstrate here is T is in NP halt; that means what I will do, I will reduce a language, which is already established as NP complete to this language T, so that, you know, I can say this is NP halt. Of course, I will reduce in polynomial time.

(Refer Slide Time: 08:53)

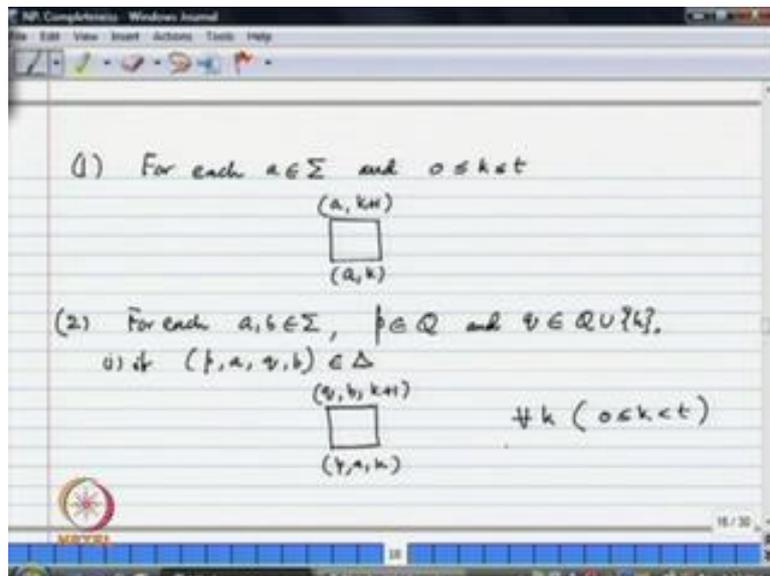


In fact, I choose this language N 1, that we have discussed in the last lecture. That N 1 can be reduced to T in polynomial time, that is what is our, my target here? So, that means, what I have to do, I give you s, if you consider a string in N 1, whatever that f x that I give, that would has to go to t and vice versa, such a reduction I have to give and that reduction should be possible in polynomial time. So, now you take an arbitrary string x in 0, 1 at the rate star. Of course, as I had mentioned, you can encode this tiling system as a sequence of 0s and ones.

And thus, I am just writing here also 0, 1 at the rate symbol. But, of course, if you have some other symbols that you are using to encode this tiling system, then appropriate alphabet will come here. So, now you take a string  $x$  in 0, 1 at the rates over this alphabet and, if it is not of the form, bracket  $M$  at the rate  $1 t$ , that mean for some non deterministic machine  $M$ . If it is not of this form, then I simply give some string, which cannot go in  $t$ . You see, as I had mentioned in the previous, this thing, when we are work, when I was giving hints for  $N 1$ , you see, when I am taking  $1$  here, is that cannot be in  $t$  at all, because in  $t$ , you see at least  $1$  at the rate symbol, that we have, so that, this is, this portion anyway is not in  $t$  at all.

Now, if the string  $x$  is of the form, bracket  $M$ ,  $1 t$  at the rate  $1 t$ , for some non deterministic  $M$ , what I would give you. I would give a tiling system this  $D M$ , because I am calling  $D$  subscript  $M$ , because we are constructing with respect to the given non deterministic machine  $M$ ,  $D M$ . And, I give you say  $1 t$  plus  $2$ . So, how this tiling size is coming will discuss now. If it is a non deterministic machine, you simply give like this. Now, what is the tiling system, that is the question, what do we suggest is, this if non deterministic machine is  $Q, \sigma, \delta, q$  naught.

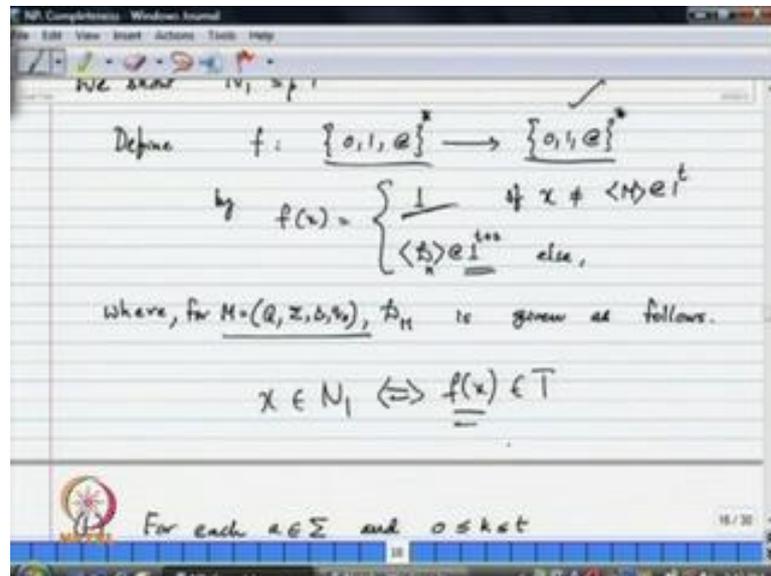
(Refer Slide Time: 11:21)



Then the tiling system, what do we do, you remember we have created tiles, when we are showing the tiling problem is undecidable. So, here also, a similar manner we create

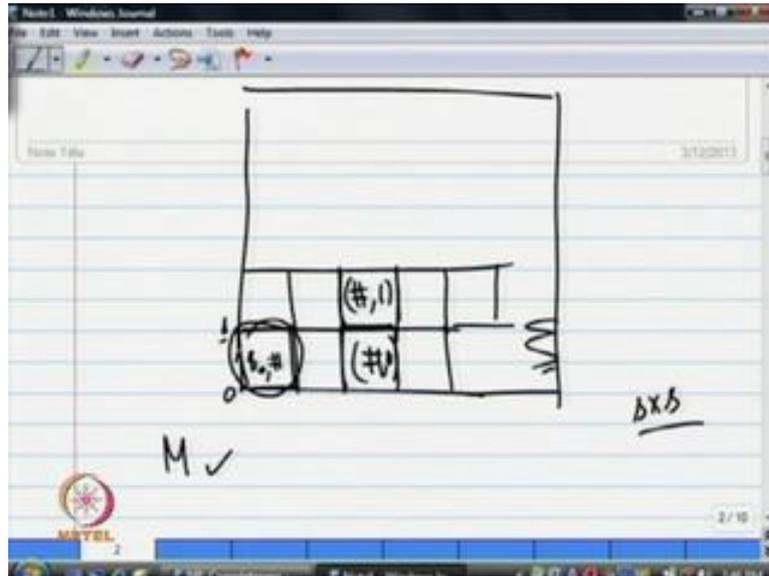
tiles, and see, whenever you have an  $s$  by  $s$  tiling, then the concerning this string is in language  $N_1$  and vice versa.

(Refer Slide Time: 11:51)



That is, how we have to create to show this is a reduction, so that means, what I have to observe here,  $x$  is in  $N_1$ , if and only if, you know,  $f(x)$  is in  $T$ ; that means, whatever the tiling system we have created, that should have you know, if I say  $1t + 2$ , that must tiling should be there, for this system and vice versa. Now, the tiling system what we are creating, I give you tiles, because the, let me first explain you, the philosophy, that how I will proceed, that is exactly the similar manner that we have considered for undecidability of tiling problem.

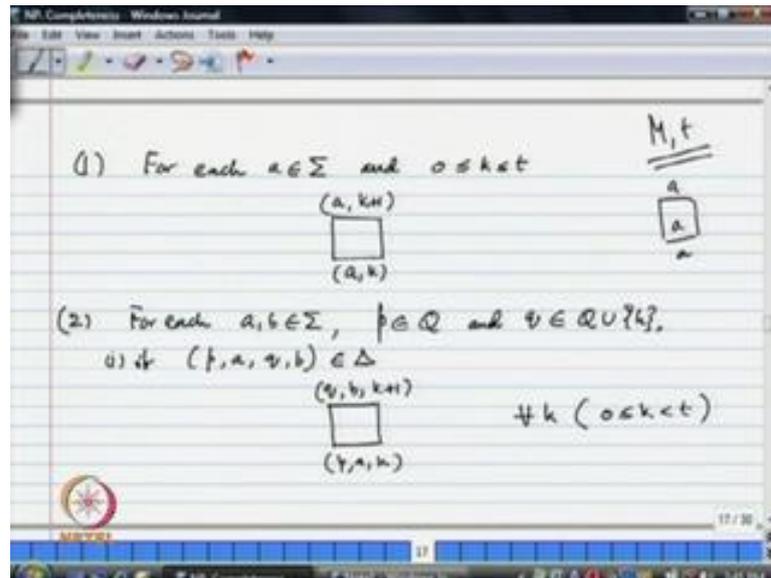
(Refer Slide Time: 12:26)



We will have the tiles, as the tape cells of your non deterministic machine, that is the how we have  $n$  says in the in that problem. And, earlier what was the situation, whatever is the tape content, that you are showing and of course, to start with the initial state, because on the blank tape, you will be starting. And, whatever is the non deterministic machine  $M$  pursues, because earlier in that case, you know, there you are having a deterministic machine, in case of tiling problem. In this bounded tiling problem, I am having a non deterministic machine, whatever is the non deterministic machine, that it pursues to halt you know, within the given number of steps, I will try to have here on  $s$  by  $s$  tiling, an appropriate with an appropriate  $s$ .

So, what are the tiles over there, the tiles are essentially the cells, tape cells in the tiling problem. Here also, what do I do, what are that cell content with respect to that, each cell will be seen as a tile and I define tiles appropriately? And, each of these rows with, what is the information of the tape that will be propagated, over the computation of  $M$ , to fill an appropriate size, you know of size  $s$ . So, this same philosophy will continue here.

(Refer Slide Time: 13:58)



So, in that connection, I will consider, tiles of this form, the first type, for each symbol  $a$  in  $\Sigma$  and  $0$ , I mean for each  $k$  up to  $t$ , because  $t$  is given to us, because we are given  $M$  and  $t$ . So, these two parameters are with us. So, I consider is it, I will have a tile for which you know, which can be fixed with the information,  $a, k$  and the same information will be there in the next level also, next row also. That is how; that means, these are the tiles, which can be used to fix the current information in a particular cell, if it is not changing.

For example, I have here say blank, and this is the  $0$ th row, because, so now this is  $0, 1$  like this. So, that means, what I am trying to say, this is the situation with this cell and if there is no change, in the second, what is called configuration are the tape content, then I will have the same this thing. But, now the situation is, the configuration, this is the first configuration second configuration so, I am representing it this by this. So, that means, this tile now, you know, the horizontal sorry vertical constraints will also be shown, will be shown by this.

So, for this tile, when  $a, k$  is the information at this, and for this tile you can have  $a, k$  plus  $1$  as the information, when there is no change. So, what is the difference, earlier you would have considered just tiles, the information need to be continued, may be, I would have written this  $a$ . So, what I am saying is here. So, instead of writing inside, I am just writing outside. So, when the information, there is no change in the information, from

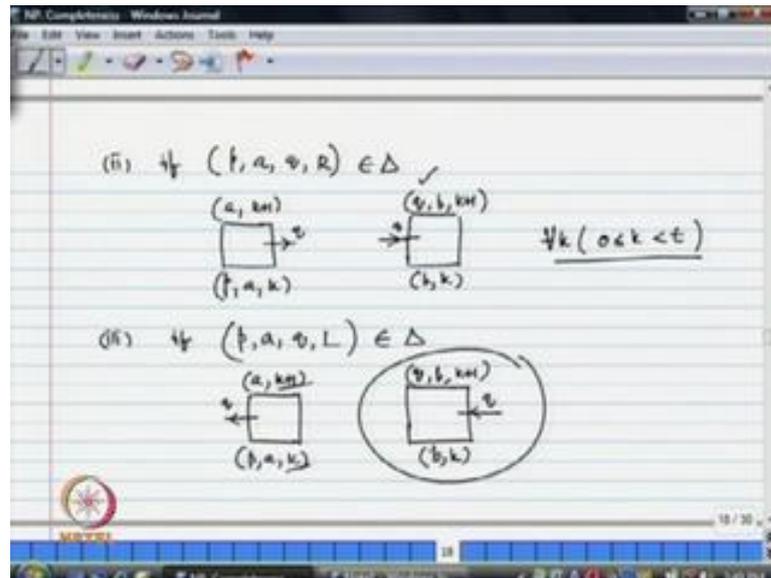
the previous configuration to this in a particular cell, that can be you know, this type of tiles can be used.

And then, regarding this transition, concerning each transition you know, the respective things need to be sort of considered. So, the transition will, transition in a particular cell will be you know, earlier say  $q$  naught blank. And now, when you are applying a transition on this, because the state component is here. So, the state component is an additional component to the particular concerning tile.

So, look at, if we have a transition  $p, a, q, b$ , because this is a non deterministic machine, this is the delta transition set. So, for a state  $p$  and  $q$ , that can be a halting state for symbols, for input for symbols in  $\Sigma$ ,  $a, b$ , if I have this transition, what do I say, I will consider, this kind of tiles. So,  $p, a, k$  you know, which can be transformed to, because  $b$  is getting printed. So, changing to the state  $q$ , you are in the current cell only, but you know, because of this transition, you will be moving from one configuration to its next configuration; that means, in the computation one step, that you have progressed.

So, this  $k$ , the parameter as I had mentioned, this gives the progression of your computation. So, if you are in the  $k$  th configuration, the  $k$  plus 1 configuration, you will have the current cell  $q$  and the current tape content is  $b$ . This kind of thing, I will consider for all the possibilities up to the  $t$  number of steps. So, for all  $k$ ,  $0$  less than or equal to  $k$  less than  $t$ , concerning this transition.

(Refer Slide Time: 17:36)



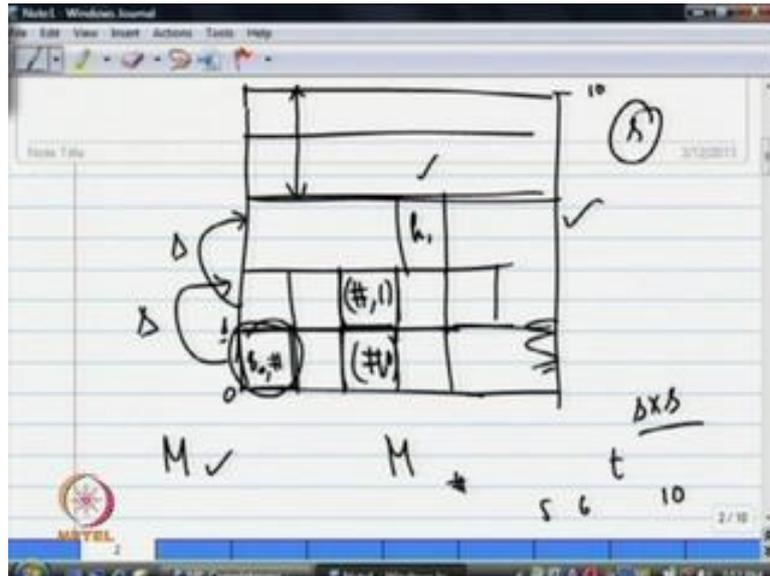
Now, if the transition is taking a right move as earlier, in case of tiling problem, we are considering, look at here. So,  $p, a, k$ ; that means, that the tape content  $a$ , from in the state  $p$ , in the  $k$ th competition, the  $k$ th step. Since, it is changing to the state  $q$  and taking a right move, though the current cell tape content is  $a$  only, in the next step. Of course, I will have a now, a horizontal constraint here, which is which this can be fixed you know, next to, because this  $q$  is going out and  $q$  is coming in, so that you know, the current cell can be updated by the state  $q$ .

So, this kind of transition you know, whenever there is a right transition, right move. Then, we consider this kind of symbol, it is exactly, what we have discussed in case of tiling problem, but what is the difference here. The difference is, the time parameter, the step parameter that we are including in this; that means, if you are in case step, if a particular configuration is there, in  $k$  plus 1 step in that particular cell, what is getting updated.

So, this is the business that we are updating here. And, now I consider such tiles which can be fixed, up to the you know,  $t$  steps. So, for all  $k$ ,  $0$  less than or equal to  $k$  less than  $t$ . Because, I am allowed to go for  $t$  number of steps with  $n$ , so, that is why, up to  $t$ , we are considering. Similarly, for left transition, a transition which takes left move.  $P, a, q, L$ , if this is the transition, like as earlier, only parameter is, only a parameter, this step parameter, that we are putting it here.

So, your sincere changing to the state  $q$ , while I am taking, this can be fixed to right of a cell, which is of this form. So, if a current content of this cell is  $b$ ,  $k$  in  $k$ th step, if it is  $b$ , now the current, the cell content is still  $b$  in  $k$  plus 1 step, where as the current state, the state component will show up in this particular cell. So, in this cell, the reading and writing head will be there.

(Refer Slide Time: 19:56)



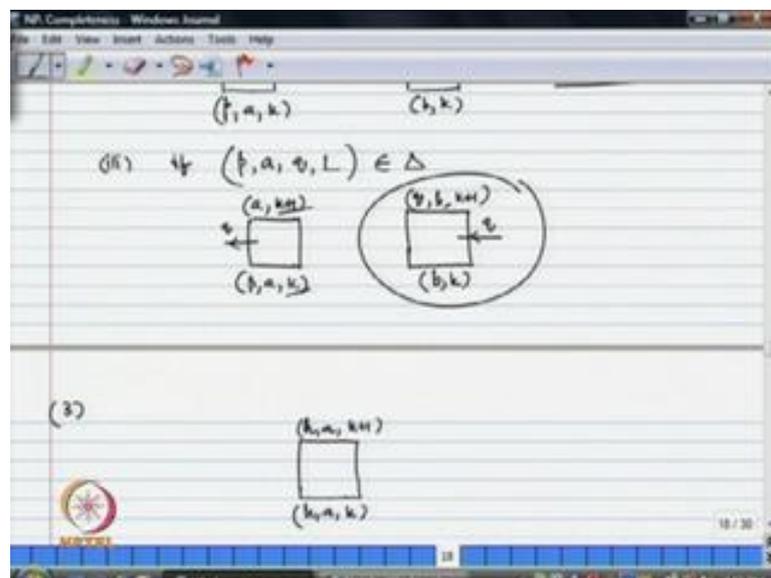
So, representing that I have, we are constructing this kind of, what is called, tile. Now, though thing is, when I am trying to make an  $s$  by  $s$  tiling of, I will go up to  $t$  steps. So, in between say for example, I come, I get a halting state. Since, I am targeting to give an  $s$  by  $s$  tiling for some  $s$ , because this turing machine you know, when you are starting on the blank tape, it will halt with in  $t$  steps, with in  $t$  steps means you know, the number of configuration that you get here or within that you know,  $t$  plus 1 number.

Now, the point is how many, how many steps that can be you know, if  $t$  is 10, if this must be halting in 5 steps, it may be in 6 steps, it may be in 7 steps, whatever it is. For example, you know, if this is 10 All right, so this is, till this point. Now, it is we are allowed you know, if the machine will halt within ten steps. So, it may be after three steps, it may be after four steps, for example, if it is halting after four steps. Now, we should still have a propagation of this configurations, you know, that computations should be shown up, so that, you will have an appropriate tiling in any case.

So, what we give here, because see the propagation is possible with respect to transition only, because this is one configuration, how do I get a next configuration, if there is a transition. So, the application of some element in delta. Similarly, if have this configuration, I can go to this, if I have some, what is called, transition. Once I get halting state, now no transition is applicable, you know that.

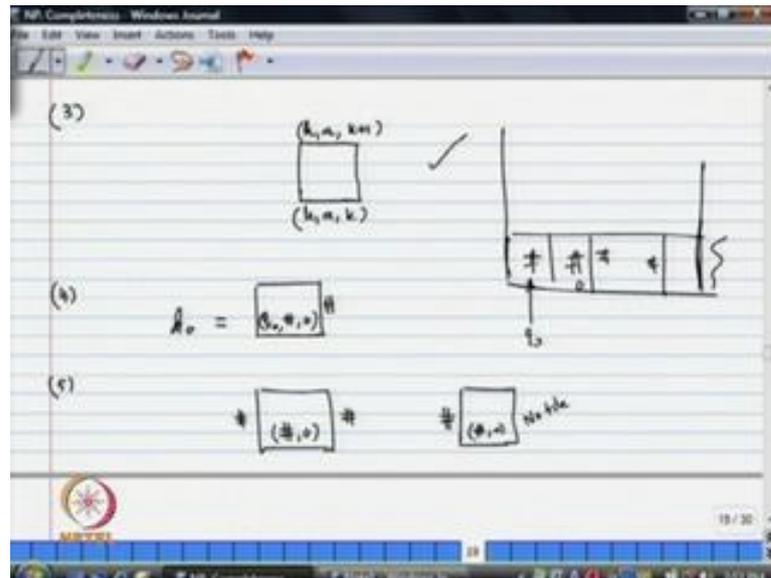
Now, how do I, then fill all these portion, I cannot fill, if I just rely on the transitions only. Now, what do I do, if it is the halting configuration, I allow you to you know, give the propagation of this tape cell, I mean, what is called tiling, with halting configuration up to a certain mark, what is that mark, that we look at, fine.

(Refer Slide Time: 21:58)



So, we look at an example and then, evaluate like, how many such tiles need to given. What did you, what this exactly indicates, if you are having the tape content  $k$ , in the case  $t$  with the halting state  $h$ , you are allowed to propagate this, what is called computation, Of course, you are not going to get any change. The same configuration that I would be writing so many times, a fixed number of times, to propagate this tiling up to, you know some  $s$ , whatever that we are targeting to show.

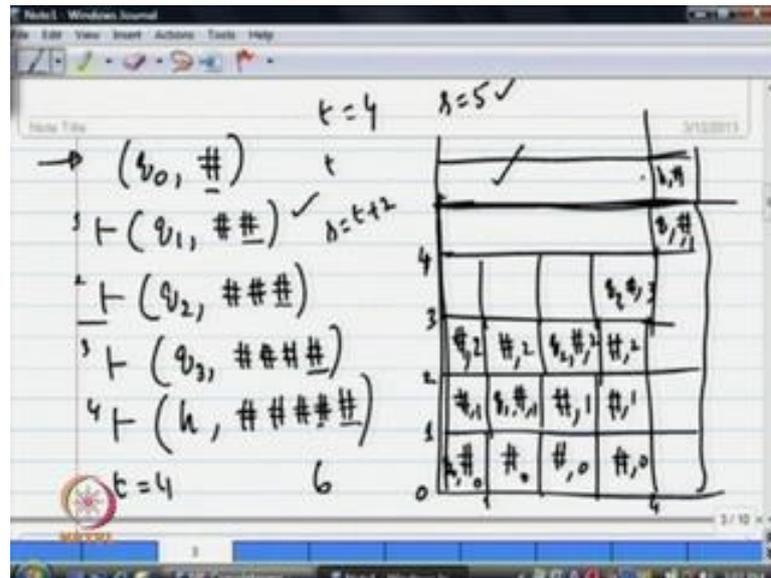
(Refer Slide Time: 22:31)



Intention of this tile is clear. And now you look at left bottom corner, this is how, this is the special tile that we should have. Because, in the left bottom corner, with the blank cell is the current initial state  $q_{\text{naught}}$ , that is you know, the 0th configuration. Hence, I say that, after first step, I may get 1 and so on. So, this kind of initial tile you should have. And then, to fix in the first, this thing, if we have  $q_{\text{naught}}$  like this, the next cell, blank and so on; the respective number of blank cells, because these will be seen as tile into right later on.

So, that way, these are all blank cells, you would require. So, blank in the 0th, these thing. So, blank 0, left to right, this is the horizontal constraints, left side is the blank cell, right side also blank cell and so on. Of course, since, we are targeting to have this bounded tiling, the last tile here, the left to that, you know, I will have blank cell and what is right to that, I am not giving any constraint here. So, this kind of tile with 0th position, I will be fixing. So, this kind of tile, we may consider, fine.

(Refer Slide Time: 24:02)



Now, let us just look at an example to talk about, how much should be the, as with respect to  $t$ . Now, if I consider a Turing machine just behaves like this; that means, how do I give. So, this non-deterministic machine, I will just give in the deterministic manner, does not matter. Assume that, when you start, it in  $q$  naught, it will go to you know, state say  $q_1$  and take a right move. And in  $q_1$ , when it reads blank, it takes a right move and come to  $q$  naught, and assumes this thus; let me just, because I wanted to halt it infinite number of steps say  $q_1$ .

And, it takes say, another right move in  $q_2$ , and another move say in  $q_3$ , it is not changing any tape content. So, the blank is that, and in  $q_3$ , in blank cell, it takes another right move and halt, for example, takes a right move. Let me, consider, the Turing machine behaves like this; that means, it simply takes you know, so many right moves and then halts, finite number of items. So, what is my, you know computation in  $q$  naught, you start with a blank tape. In the next step, it will go  $q_1$  and takes a right move, All right, so that means, that is the next configuration.

And, in the next step, it goes to  $q_2$  and takes another right move, so that, this is the configuration. Then, next step, it will go to  $q_3$  and takes another right move. So, this is the next configuration and then, so, now it suppose that halts. So, it takes a right move and halt suppose now. So, how many steps it took, step one; this is step two, step three, step four steps. Say for example, someone is telling me, that this halts in say six steps.

Now, what exactly, I wanted to look at here is, say this blank, this state component  $q$  naught, this is in that 0<sup>th</sup> step. So, other cells are blank, blank in the 0<sup>th</sup> cell, so let me put 0 like this. And, in the next configuration, what I have here. So, the tape content is blank that is not getting change. So, it is taking a right move. So, this  $q_1$  goes here, this is blank. Now, this is you note, the first after you know in the first step, so I write 1 here, now 1 here.

And, then this propagates, this is the situation and so on. Now, in the next step, this is the tape cell content is this, in the  $k$  step, I have 1 here; in the  $k + 1$  step, that is 2. And then, this is again, this  $q_1$  and the blank, then you read, you takes a right move and goes to  $q_2$ . So, since it is taking a right move, the tape content is this in the next step, but here, you have  $q_2$ , this blank in the second step All right.

So, and of course, here is a blank in 0 steps, and then, this will be blank in a first step, and there is no change, blank in the second step likewise, we have. And, then we will have here, say  $q_3$ , blank in the third step. So, likewise, we have the propagation of tiles that we wanted to fix. And then, in  $q_3$ , I will get say halting in the next stage. So, this is the halting, the tape content blank in the fourth four steps. Since, someone told me that, this halts in 6 steps, so that,  $t$  for me is six.

So, and since, it is halting in four steps, I know this much tiling is possible, but since, we are targeting with respect to  $t$ , and it may be three steps; it may be four steps; it may be five steps, what I should have, I should have a mechanism of propagating this tiles up to, you know, some number with respect to that  $t$ . Thus, what we suggest is, you have tiles with halt with halting state, that you can propagate up to so many.

Now, you can ask me, what is the minimum number that I should fix for this tiling, what is that size, now you look at. Here, if it is halting in exactly four steps, we look at, this will be the first row, because that is 0 and this is coming in 1; this is coming in 2; this is coming in 3; this coming in 4, this is coming in 4. Now, since it is halting may be, this is the best that you can stop. So, this is halting in  $t$  equal to 4 steps, now what is the tiling that we have here, look at this size of this, what is the size of this, that is you know, 0 1 2 3 4, so within four steps.

So, this configuration is fixed here, and then this configuration is coming here, two, three four. So, at this point, this will halt. So, within you know this, we have four steps here.

So, this much tiling can be done here. Now, what is an appropriate, what is this tiling here, so, how many cells that we have filled here, so, 0 1, 1 1, 2 2 and till this is 4. So, if I say, if it halts in four steps. So, we have thing like this, the halting configuration has come here. Now, for example, if you take anything  $t$ , little, little more than any  $s$ , which is little more than this much, because within these many steps, this will halt.

So, even if it halts, if I take little more than this, I can actually continue to propagate through halting state and make a appropriate tiling, that is what is the target. Now, what do I choose here is,  $s$  is equal to  $t$  plus 2. So, now, if you take  $s$  is equal to  $t$  plus 2, this is an optimal number, we can look at, why this is, because after the halting configuration, if you come, if you get, you can propagate at least one more time.

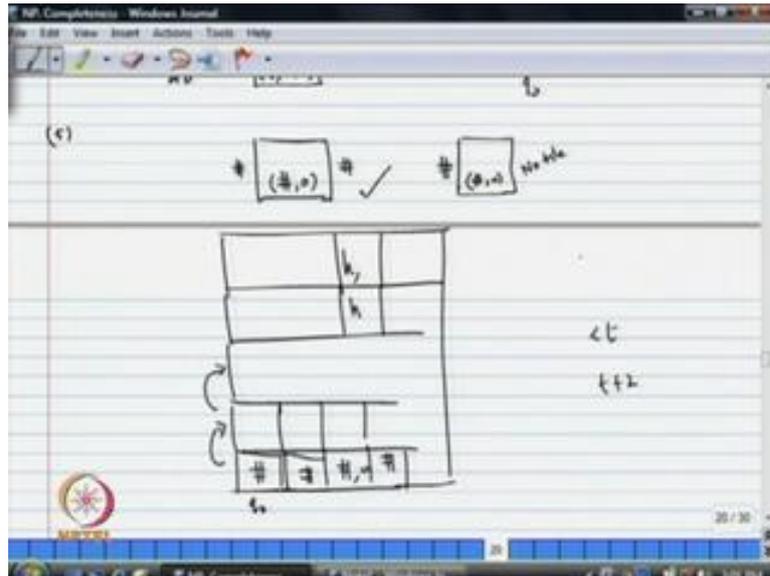
And thereafter, you will be, if you want, if you required, you can continue. So, this gives you at least one more time, I can propagate with the halting this thing. Once I realize that, I have, I am started propagating with halting only, then is the, this has come to halting state. For example, if I just fix at this boundary, if this is not halting state, suppose if I say, I want to do just, I have only say,  $s$  is equal to 5. Then, this is what is five by five tiling, because 0 1 2 3 4, now this entire thing is filled, suppose this is not halting state, then what we what will be the situation, you have got already  $s$  by  $s$  tiling,  $s$  is equal to 5.

Now, I wanted to leave at least one more, this thing blank, if I do not get within these many steps halting state; that means, say for example, here instead of  $h$ , if I have, say for example, some  $q$ , some state  $q$ . If I just declare this much only; that means,  $s$  equal to  $t$  plus 1, because this is five by five tiling. Five by five tiling is from 0 1 to 4 cross 0 1 to 4, if I just take  $s$  by  $s$  tiling, with  $s$  is equal to five in this case, even if it is not halting state, then the you have got that tiling, complete tiling.

So, what I wanted to look at, if I take one more; that means,  $t$  is equal to sorry  $s$  is equal to  $t$  plus 2, within  $t$  steps, if it is you know, within  $t$  steps, if it is halting with this. Then, what we are going to have, once I realize that, halting state is there, then I am able to propagate it here. And, if it is not halting, what you see, if it is not halting with this, then you will have the propagation always, but you see, we are we have not defined such tiles here. Because, we have defined tiles to take the transitions up to, you know the steps, which are less than  $t$ .

So, I will not have tiles, if it is not halting, whenever it comes to the halting state, then I am, I can propagate till, whatever is the appropriate number that I have. So, we will give this kind of tiling for that. So, k we will fix. So, now, if I take, s is equal to t plus 2, once I reach to halting state, then I have a mechanism to propagate using halting state further and give the tiling.

(Refer Slide Time: 34:22)

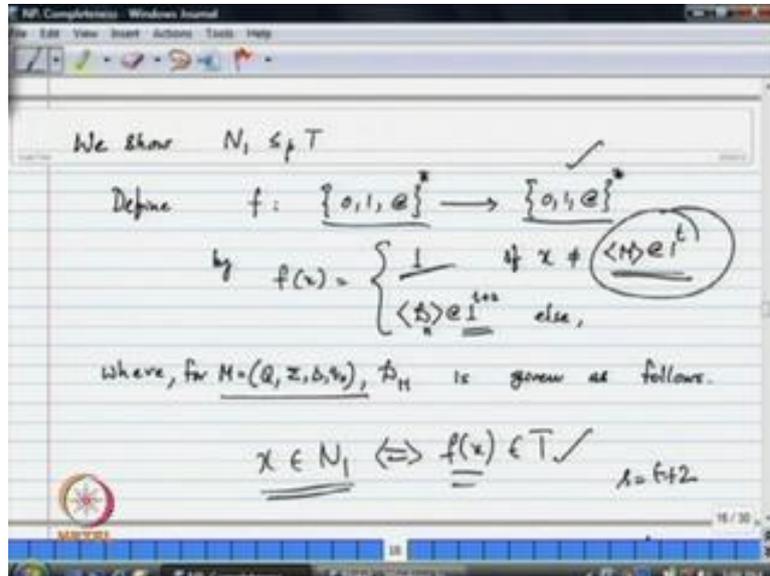


So, let me give a note on it, let us look at. So, in the, at the first place, the tile you know, we are fixing with the blank and q naught. And then, you are fixing all these, are blank cells, so this is, what is the first configuration? And, in the second configuration, whatever is the state updated. So, second configuration will be given appropriate tiling like this and so on, whenever it is halting with within t steps, so that means, when, I get a halting state here. Because, these transitions will, you will be giving with till less than t, this kind of tiles. And, once I get a halting configuration, then up to t plus 2 steps you know, we will have, we will be able to propagate.

So, this is what one more step, we are considering and we have already given tiles of this form, so that, you know, in 0, you will be starting with all these information, with all these blank. And then, you update the transition, you will be able to update the transitions up to a certain mark; that means, up to t plus 1 by t plus 1 tiling, you will be able do, with respect to the number of transitions, if it is not halting. After that, these two particular things, you will not be able do, because, so till this point you can go, if you

have halting state here, then you will be able to propagate to this, so this you can go, k less than or equal to t.

(Refer Slide Time: 35:53)



Example, if we go. Then, you can recommend to have you know, t plus 2 by t plus 2 tiling with the system, now you understand. Given it non deterministic machine M and if it halts within t steps; that means, if it is N 1, now these tiles, because the configurations are essentially shown as tiles. So, in this tiling system, it will have s s t plus 2 by t plus 2, s is equal to t plus 2, s by s tiling, you will have, so that, that will be in p. Now, in this tiling system, if you have an s by s tiling that is s is equal to t plus 2, what is the meaning of that?

Because, what are the tiles that we have created, up to t by t plus 1 by t plus 2 t plus 1 tiling, if the machine is not halting anyway you will get. But, you will not be able to get the last row; that means the t plus 1 step, that particular row, you are not going to get, because we have not given tiles for that purpose.

So, if the machine is not halting, then you will not be able to and if the machine halts, before that many steps, I get a halting state, and I will be able to propagate that and so that, I will have a, s by tiling. So, given M, if it is not halting, I mean when, I do not have s by s tiling; that means, there will be at least one row free. And thus, the corresponding x is not in N 1, if I have a tiling; that means, the machine is halting within t steps, so that, the string x is in N 1, this is construction clear.

So, I will just recap once again, what we are trying to do here is, we are trying to reduce  $N$  to  $t$  in polynomial time. So, this reduction, how are we doing, take a string, if it is of the form  $M$ , this non deterministic machine, machine  $M$  and at the rate  $1/t$ , we are trying to give a, we will give a tiling system. This will have  $(t+2) \times (t+2)$  tiling and vice versa, that is how we are producing. So, what are the configurations that you have; that means, the tape cell that we have.

Each cell may be treated as a tile, as in case of tiling problem and each of these configuration can be seen as, the tape can be seen as, each cell as tile and the configuration that we are propagating, will be seen as a tiling. Now, if this string  $x$  is in  $N$ ; that means, the non deterministic machine  $M$  halts within  $t$  steps. Then, you know, I, when once I fix that is  $(t+2) \times (t+2)$  within this  $t$  steps, I will be reaching the halting configuration and with the state  $h$  and we are giving a tile, which can be propagate in the halting state, up to one more or and after the steps up to the tiling  $(t+2) \times (t+2)$  tiling, All right.

And, if the machine is not halting; that means, you will have infinite propagation. So, at the stage you know, for  $(t+1) \times (t+1)$  tiling, you will get the  $(t+1) \times (t+1)$  tiling, but the last row, you cannot get, because you do not have tile, which can propagate up to  $(t+2) \times (t+2)$  size, because all these styles, this style, this style, these are all given with further step up to  $t$ .

So, this is also given up to  $t$  and with all these vertical horizontal constraints, but whereas, this I will give to fill up to one more, because of to the halting state. Thus, we can have this kind of, this, what is called, tiling, if and only if,  $x$  is in  $N$  alike. And now, what you have to observe, that this kind of construction is possible in polynomial time with respect to the input size. Because, what are the tiles we are constructing here, because  $M$  is encoded here, that is part of the input and you see, how many tiles, that we are giving.

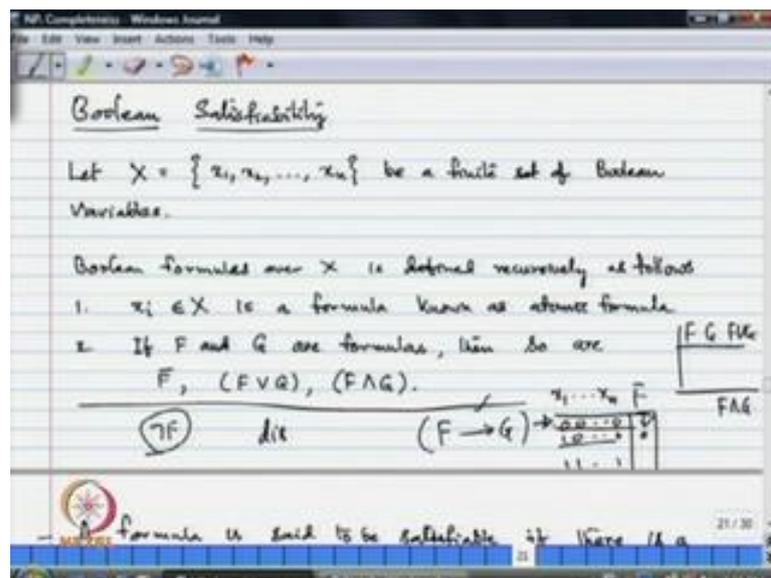
So, with respect to each symbol and this, we are giving these kind of tiles and with respect to the number of transitions, we are giving these kind of tiles, for every  $a, b$  in  $\Sigma$  and all these. So, again, the number of transitions and the number of symbols are  $\Sigma$  and this  $k$  you know, what are that we are choosing, these are all finitely many.

So, you will have finitely many tiles with respect to the input parameter. And similarly, these cases, these are all finitely many tiles.

So, these tiling system is you know, is in polynomial with respect to the given input size. And, you can argue now that, what is called, the tiling system can be created in polynomial time with respect to the given input parameter, so that, you can observe that  $N \leq P$ . So, that bounded tiling problem is NP complete. So, now what I will do, I will give some more NP complete problems in this line up, and we can nicely cross check the and learn like, how to show given problem is NP complete.

There are several variants as I had mentioned of this NP complete, we have talked about NP and many other and we have reduced NP to this tiling problem. Now, I will as, I had mentioned, in this lectures, I will also give, you know the, first NP complete problem, which is established, so called, the satisfiability problem, that was the first NP complete problem, it was established by Stevin Cook. So, the Cook's theorem says that, satisfiability problem is NP complete. So, I will give a proof for Cook's theorem in these lectures. First, I will talk about the problem, I will introduce the problem and then, I will say like, how, what is the approach that we would have, to show that, this is NP complete.

(Refer Slide Time: 42:11)



So, the Boolean satisfiability, you all know about these Boolean formulae, but let me just, react write here. Let,  $x_1, x_2, \dots, x_n$  be a finite set of Boolean variables, because you

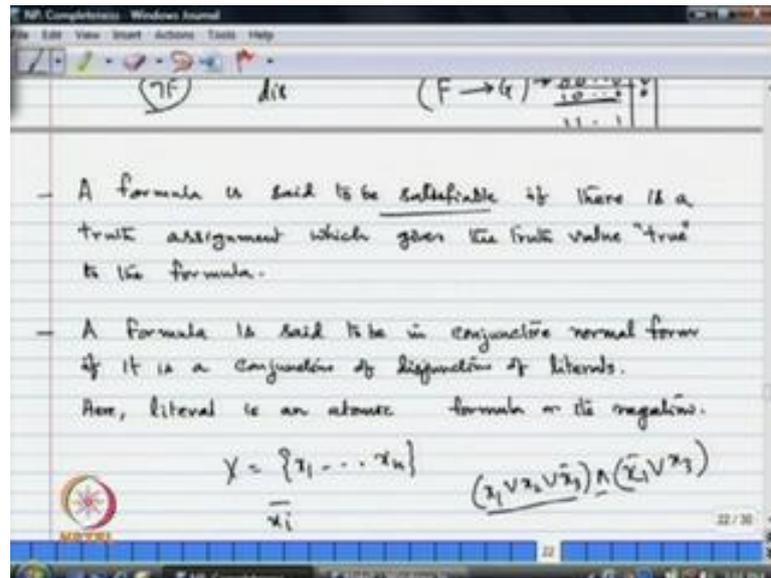
know, Boolean formulae, there defined over a set of the finite set of variables. So, using following rules recursively, Boolean formulae over  $x$  is defined recursively as follows. Each  $x_i$  from the set is a formula, this is known as atomic formula. And, if  $F$  and  $G$  are formulae, then so are, this  $\bar{F}$ , this one may write negation  $F$ . This is also a Boolean formula  $F$  or  $G$ ; this is what is called disjunction.

And then, conjunction of two formulae is also a formula. And of course, triple might even say, this is also a Boolean formula, if  $F$  then  $G$ , that can be, so this connective is also included in some of the definitions. But, any way, we will consider just this much, because you also you know, all the Boolean formulae, which can be defined recursively using these two rules, so that means, any formula by finitely many applications of these two rules, whatever that you get, we call them as Boolean formulae, you want to include these kind of connectives in all there, that you can extend this definition and have the Boolean formulae.

Now, a formula, a Boolean formula is said to be satisfiable, if there is a truth assignment, which gives the truth value through to the formula. You know the truth ((Refer Time 44:01)). So, in the formulae  $F$ , the variables from  $x_1, x_2, x_n$ , whatever they are available with respect to that, you know how to give, for the formula, truth value. Because, for  $x_1, x_2, x_n$ , the respective values, when you are given, the possibilities of truth values from 0 and 1, we can have, you know two power  $n$  here, two power  $n$  combinations and with respect to that  $F$ , what is the truth value that it will receive, you know for all these things, 0, say for example, for this 1 likewise.

So, there are two power  $n$ , possible of possibilities of truth assignments and each assignment, what is the value that  $F$  is receiving. So, these are the values, say for example, 1, 0, whatever that, which we are going to get. Thus, you know for atomic formulae, whatever that you are assigning for negation of  $x, x_i$ , if you are giving 0 here, negation of  $x_i$ , you are giving 1. And similarly, for  $F$  or  $G$  you know, the truth tables like for  $F$  and  $G$ , if you know the truth values, you know for  $F$  or  $G$ , this truth table you are aware of and similarly, for  $F$  and  $G$  for conjunction.

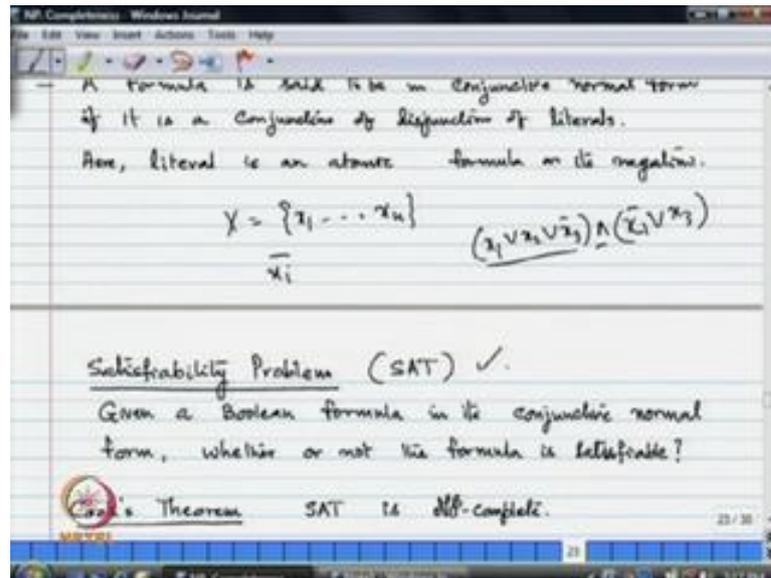
(Refer Slide Time: 45:29)



So, these truth tables are known to you, so that, you know for a given formula you know, what are the truths, what are the possibilities of truth values. Now, we say formula is said to be satisfiable, if there is a truth assignment, at least one row against to which, you should get, you know value 1 for F. A formula is said to be satisfiable, if there is a truth assignment, which gives the truth value true to the formula. Now, what is satisfiability problem we will consider, what is the variant of the satisfiability problem?

That is, I am not worried to give any arbitrary formula; I take a formula in c n f, conjunctive normal form, what is conjunctive normal form. Conjunctive normal form, of a formula, is of the form, it is conjunction of disjunction of literals. Now, what is a literal, literal is an atomic formula or its negation; that means, in this present context, what I have, x is equal to x 1, x 2, x n and now literal is either x i or negation x i. So, these are called literals. Now, a conjunction of disjunction literals, I mean in a formula should be of the form, say for example, x 1 or x 2, you know or x 3 bar and say x 1 bar or x 3 and... So, this is we see, this is a disjunction, so conjunction of disjunction of literals. So, a formula which is in this form, we call it as c n f.

(Refer Slide Time: 47:04)



Now, Satisfiability problem is given a Boolean formula in its conjunctive normal form, whether or not, the formula is satisfiable. Now, I give you any specific format, specific normal form. Now, we have to verify, we have to cross check, whether the given formula is satisfiable or not, that problem is called satisfy problem, satisfiability problem, let me write it as SAT. Now, Cook's theorem states that, this problem, satisfiability problem is NP complete, what do I suggest, now in the routine manner, because we were doing, how to establish a problem is NP.

To prove SAT is NP complete, Of course, you have to observe the SAT is in NP and then, we have to observe that, SAT is NP halt. What do I suggest, you establish SAT is in NP, because given a certificate, which in a polynomial time, you should have polynomial time decision procedure to verify that, the given formula is satisfiable or not. So, that means, essentially in our procedure, we have to identify a polynomially balanced language, which can be decided in polynomial time.

So, you do that part, as an exercise, I will in the next lecture, I will give a proof or you know, I will give a reduction from already establish NP complete languages to this satisfiability problem, so that, we can establish that SAT is NP halt. So, in this classroom, since, we have already established few languages, some of the variants of NP naught also, one can work out as classroom exercises. So, I will make use of these things

only, the two establish that, satisfiability is NP complete, to prove this Cook's theorem in the next lecture.