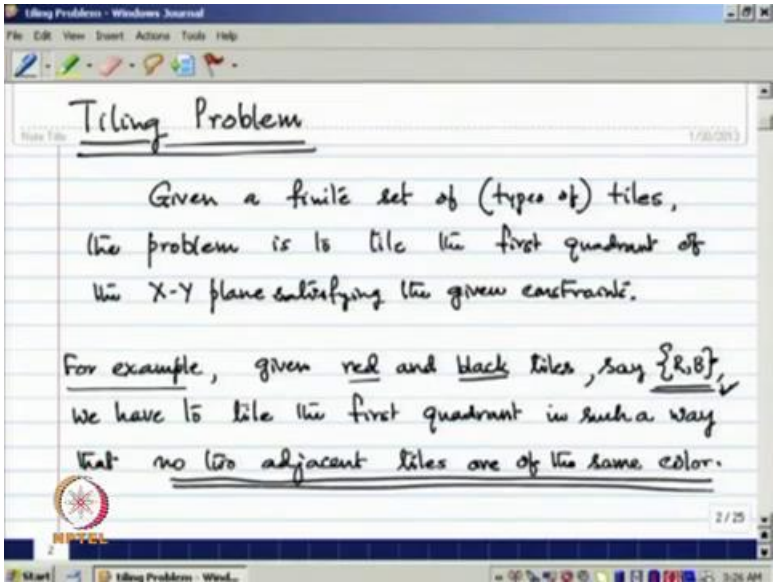
Formal Languages and Automata Theory
Prof. Diganta Goswami
Department of Computer Science and Engineering
Indian Institute of Technology, Guwahati

Module - 13
Decidability and Undecidability
Lecture - 4
Undecidability Part – 3

In continuation which are discussion on Undecidability, when we are going to introduce another problem known as tiling problem, and so that this problem is also undecidable.

(Refer Slide Time: 00:34)

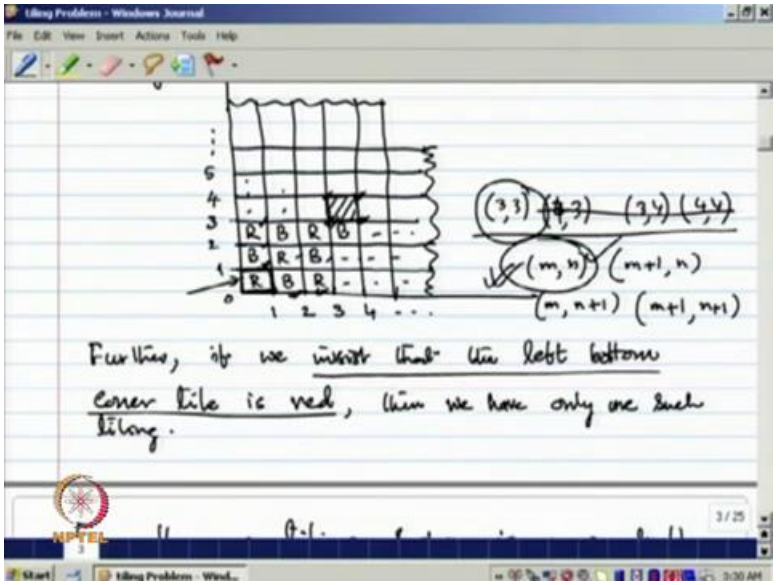


Now, the tiling problem is like this given a finite set of tiles; that means finite set of types of tiles. The problem is tile the first quadrant of the X Y plane satisfying the given constants. For example suppose given an red and black tiles, so these are types of tiles, say this set is set of two elements R and B, R represents red tiles and B represents black tiles.

So, in this case what we assume that, tiles from of this type be available infinitely, so there is infinite numbers of tiles of each of the tiles either red or black. So, given this red and black tiles we have the tiled of first quadrant in such a way that no two adjacent tiles order same color, so this is a constants. So, there is some constants satisfying this

constant we need to tile, taking tiles from this kinds this set, we will tile the first quadrant of the X-Y plane, so this is the problem, so in this example.

(Refer Slide Time: 01:53)



Now, if you consider this example so if you can have this kind of tiling, suppose the first quadrant is suppose which filled up, which R that is the red, then according to the condition, since the second one are entirely this one, can we filled by on the black tiles. So, next one can be tiled by a red tile, and so on; that means, we can place all kind of types of tiles horizontally. Similarly, above R again you can have this B there is only possibility, because we have a same color, then after B we can this R vertically, so according to vertically we can fill up like that.

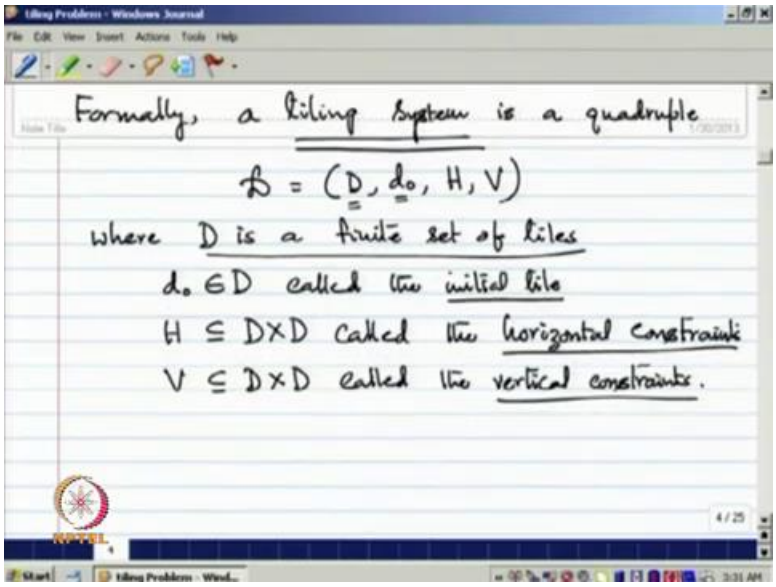
So, once this B is fixed we can fill the remaining tiles like this after B it is R, then B and so on and the condition is satisfied both horizontally, vertically. So, therefore, this is tiling; that means, that is way you can tile the whole first quadrant in the X-Y plane given infinite numbers of numbers of red and black tiles.

Now, if you insist that there is a condition is that the left bottoms central tile is red; that means, the first one in the quadrant ((Refer Time: 03:22)) of this quadrant, if it is fixed suppose it is R red tile, then there is only one solution for this problem; similarly, if it is say a let, then also we have only one solution. Now given this kind of problem, so what we can do is that suppose just consider this particular tile, so this particular tile we can be

identified as the quadrant say 3, 3 is a left bottom corner, then right bottom corner is divided 4, 3 then the top left top corner is put sorry it is 3, 4 and right top corner is 4, 4.

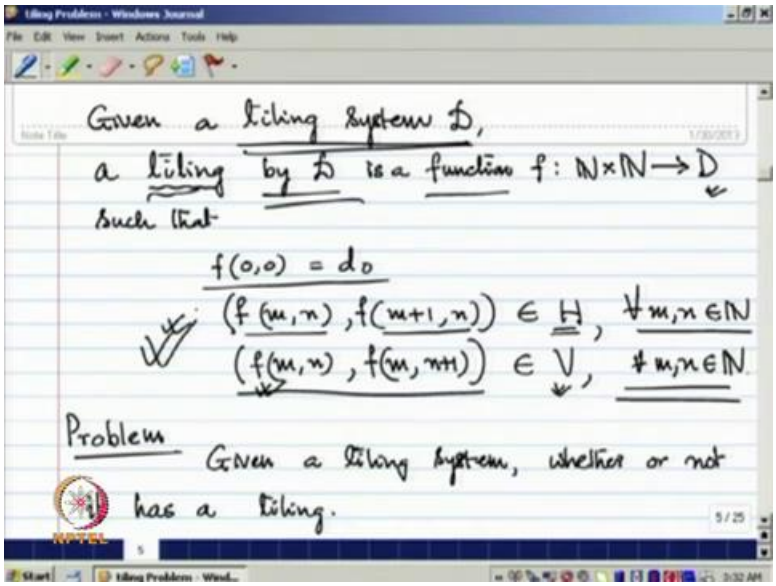
So, you can un identify this tile by giving the quadrant on the various corners, but simply if you give the quadrant of the left most, left bottom corner this one. Then, so we can identify this tile, if we have the cumbersome that we always give the code at on the left bottom corner. Then, even it is suppose m n is the quadrant at the left bottom corner, then horizontally we can keep on increasing the represents m plus 1, m will be the right bottom corner; similarly m, n plus 1 will be the left top corner and m plus 1 n plus 1 will be right top corner. So, that way if you give the quadrant of the left bottom corner it is sufficient to identify a particular tile, so although to identify a tile by giving the left bottom quadrant will be left bottom corner.

(Refer Slide Time: 06:03)



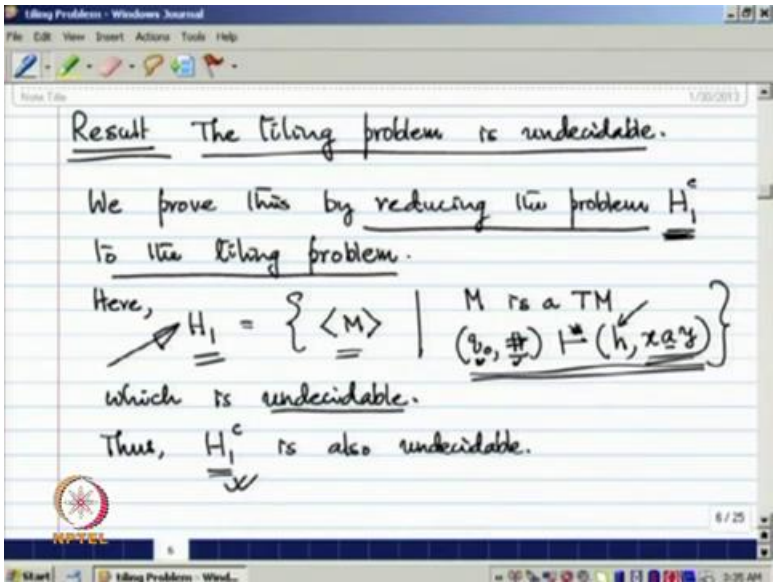
Now, once we have that comprehension formally we can define tiling system like this, so formally a tiling system is a quadruple having four elements D, where D is a finite set of tiles; that means, the type of tiles, then d 0 it is called initial tile, then is a condition which tile should be used to put the quadrant 0 0. Then, H is the subset of D cross D is called horizontal constraints, so it has to satisfy that constraint and; similarly, V is also a subset of D cross D called vertical constraints.

(Refer Slide Time: 06:54)



So, once you have this tiling system, Suppose the tiling system is D, a tiling by D is a function from N cross N to D where D is the types of tiles such that f (0, 0) is d 0 that is the first condition. That means, which tiles should be used on the quadrant 0 0, then this is the horizontal condition f (m n) and f m plus 1, n; that means, once you fix consider this tile in the location quadrant m n, then the tile which will be D horizontally the next one it must satisfy the constraint given in this tiling system for all m n in N. Similarly, this represents a vertical condition if we fix the tile m n and m n plus1it is which satisfy the vertical condition given in V for all m, n. So, given a tile system a tiling by d the is a function satisfying in all this constraints. Now, the problem is that given a tiling system whether or not it has tiling, so that is the problem, what do you want to show is that.

(Refer Slide Time: 08:24)



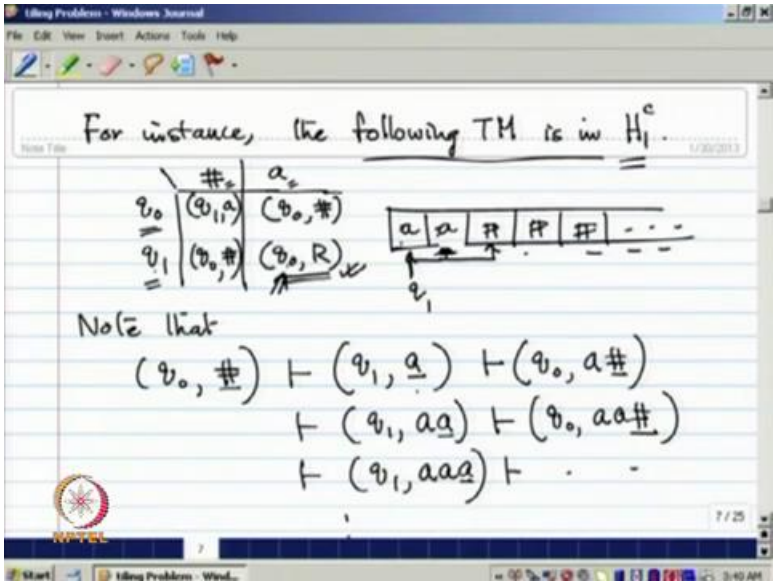
The tiling problem is undecidable, for any given tiling the tiling problem that is the given tiling whether or not it has this is tiling this problem is undecidable. To prove we use reduction from a variant of the halting problem to problem this by using the problem compliment of H 1 we define H 1, now we reduce the compliment of H 1 to the tiling problem. Since, H 1 is undesirable H 1 compliment is undesirable and therefore, this tiling problem is also be undecidable according the reduction.

Now, how will define H 1 the variant, so H 1 is basically, or a language which one are simply H 1 is set of all Turing machines M such that Turing machine starting with a blank tape. So, it is starts at starts at q 0 and it initially the tape is empty star is a blank tape, and eventually in enters in the halt state this is the halting state H and there will be some content on the string. And we can show that this column H 1 is also undecidable; that means, given a Turing machine if it is starts the blank tape over all it will eventually halt or not.

So, if H 1 is undecidable, then so is the compliment of H 1; that means, given turing machine any arbitrate Turing machine the Turing machine it does not halt, this is the problem, so this is H 1 compliment. So, we can show that H 1 is undesirable by reducing the halting problem to this variant of this halting problem, and hence H 1 compliment is also undesirable. So, leave this as an exercise to show that H 1 compliment is undecidable, now we will use this problem compliment of H 1 to prove that the tiling

problem is also undecidable we will reduce this H 1 compliment to the tiling problem to prove this.

(Refer Slide Time: 10:57)



Now, consider the following Turing machine which is in H 1 compliment; that means, this Turing machine never halts, which is halts in a blank tape and it will never halt. So, what is Turing machine does is that it has two steps q 0 and q 1 q 0 is a initial state reading a blank symbol which is say hash blank is defined by hash. So, it will enter in q 1 and prints a, so it is blank symbol become harder to a, and will send it is states to q 1, so one it is in q 1 it simply goes to the right side on your season a, and it sends its state to q 0.

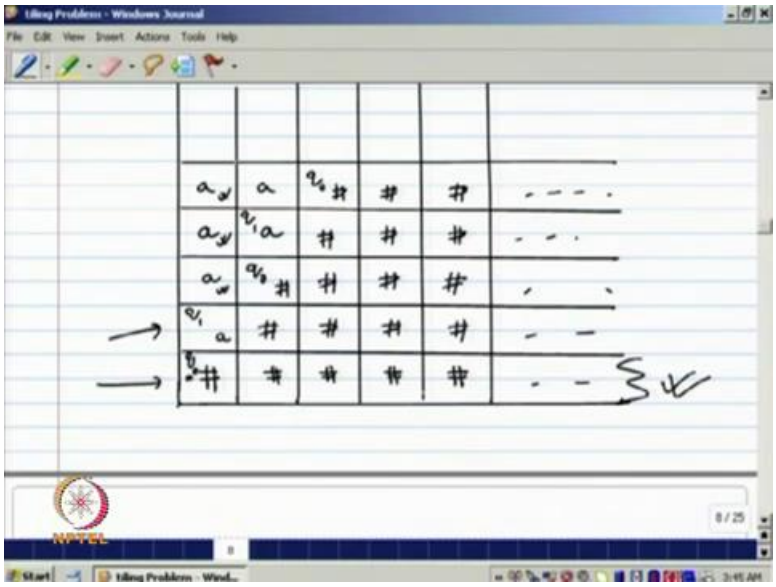
So, since total right side we have infinite numbers of blank only again in q 0 looking at blank it will print a and sends state to q 1 and q 1, since the input is a it will go to right side it will move to right side and it will sends that to q 0. So, this will continue; that means, if the content is initially say all hash is ((Refer Time 12:30)) all hash or blank and so on. So, initially this is the situation in q 1.

So, this hash will be or blank will be modified to a and it will go to this hash will be modified to a a will be printed and will sends that to q 1 and q 1 reading a symbol a you simply go to right side. And then it will sends that to q 0 q 0 again looking at hash or blank you simply print a, and it will go to state q 1 and q 1 since it is a it will again move to one side, once all to right and it will continue. So, therefore, all these hashes will be all

the blanks will be compared to a and will keep on going since we have infinitely many blanks towards the right side of the tape.

So, therefore, computation of this Turing machine will never halt; that means, q 0 the tape is head is reading blank will enter into the state q 1. So, this is the next configuration configuration q 1 a will be printed next configuration is that q 1 two i will go to q 0 and it will go one cell to the right side, so a is allowed here, and then next a is responding to the next blank, so this will continue. So, eventually q 1 a a will a responding to a and so on. Just considering this Turing machine what we can do is that, so the idea is that once Turing machine is given.

(Refer Slide Time: 14:54)

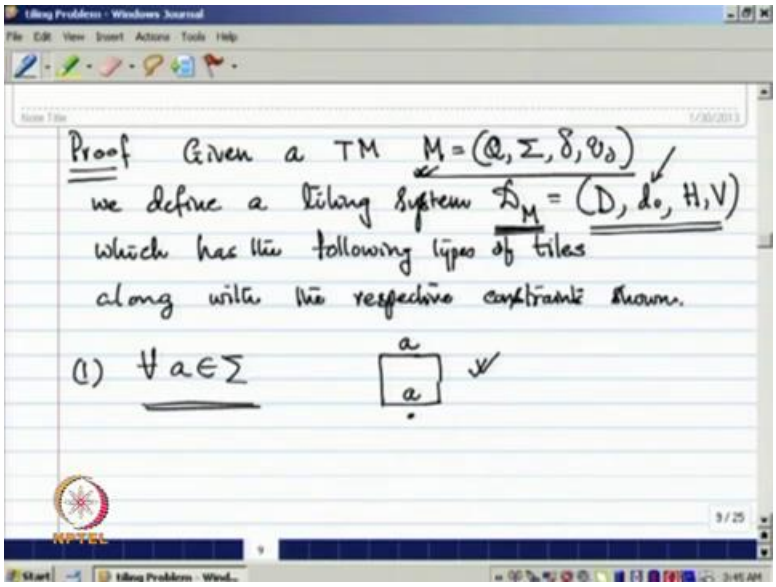


So, this Turing machine computation or successive configurations of a turing machines can be used to fill up the first quadrant by filling the one row of the first quadrant by successive configurations of the Turing machine; that means. So initially it is a q 0 hash, so initially all are blank, so you put in the first quadrant on the first row of the first quadrant this q 0 indicates that this is a start state. So, q 0 hash, and then the q 0 blank and all blank, so needs the initial configuration and you have place in the first row next q 0 on running blank it enters in state q 1 and prints a ordered and remaining are same, so therefore, we simply use the next configuration to fill up the second row next row. So, next q 1 on a will simply go to, so the head will be moved towards the right side according to our rule q 1 on a will go the right side and sends that to q 0.

So, therefore, the next configuration will be v, so q 1 on a, so this will remains same in the third quadrant, and then will enters state q 0 and this is hash blank. So, this state symbol is used in a cell to indicate that indicate a state, and that it is reading that a corresponding symbol. So, the next configuration will have you will be at, so q 0 has the q 0 blank, so this will be converted to a, and in entrance state q 1, and the remaining are same, all are blank again the next it will be a will a, so it will enters state q 0 it is blank so on.

So, this way we can keep on filling up, the rows of the first quadrant by successive configuration of the Turing machine. Since, the Turing machine never halts, so therefore, we will be able to have a tiling on the first quadrant according accordingly, so this for to use for tiling of first quadrant given a Turing machine.

(Refer Slide Time: 17:32)



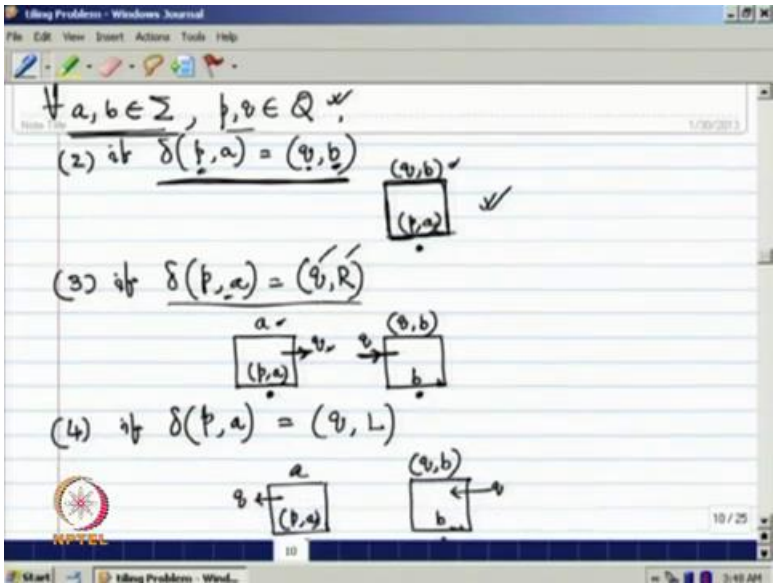
Now, let us prove this formally, so given a Turing machine M which is a quadruple Q sigma delta q 0 we define the tile system from this Turing machine. So, it is D M we emphasis that this tile system has been constructed from M the Turing machine given Turing machine M. So, our idea is that given a Turing machine M we need to have a tiling system such that Turing machine never halts Turing machine does not halt if and only if it has a tiling.

So, from M we have a we have constructed instance of tiling, so that we can have a tiling on this instance if and only if the Turing machine never halts. So, it has again for given a

Turing machine M, we define the result corresponding Turing machine for this Turing machine D M which is a quadruple having four elements to be tell D, d 0, H and V for a capital D is a set of tiles in types of tiles in types of tiles. So, these are where the initial condition is, and H and V are horizontal halt conditions which as the following types of tiles.

So, in this tiling system we will have the tiles according to the conditions of the Turing machine corresponding to every Turing machine we have some kinds of tiles. now for all a in sigma we will have this kind of tiles so what we say is that this kind of tiles is used to simply communicate any un change symbol upwards from configuration to configuration. So, it is content a and on the a is on the top a is we have the symbol a to this simply communicates that there will be some symbol which are un sense for example, (Refer time: 19:57) so in this case we have this a and we have, so we have this a, so this is a, so which are carried forward the symbol a. So, accordingly we will be using this kind of tiles, and here we have used the duct over here to just to indicate that to use this kind of tile you must have some tile below it, if you want to use this tile some tiles below it must be already available.

(Refer Slide Time: 20:36)



Then, for all (a, b) in sigma, for all (p, q) in the state of the Turing machine, if we have a transition like this in the Turing machine delta p, a equal to (q, b) it says that we have (p, a) as a element (p, a), so (p, a) will be there in the tile is contended tile on the top pest it

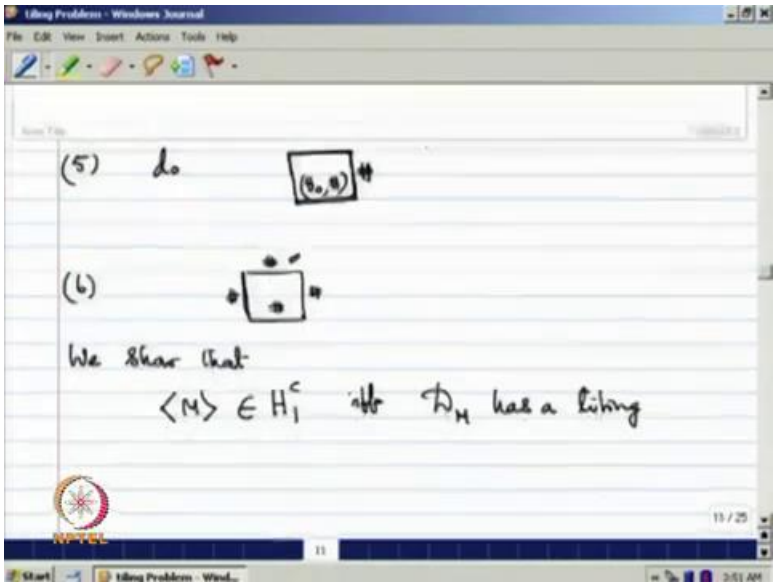
is (q, v) communicates the head position upwards, and also sends how the symbol, and state appropriately; that means, from tile below tile upward the head position is sent and state is send from p to q and the symbol is sends from a to b.

So, that happens whenever you have delta (p, a) equal to (q, b) we have a state sense from p d q and we print a new symbol instead of a and that is your b instead of a we print a, b, so that is how we communicate over here. Again, to use this kind of tiles we need to have a tile below it that indicate by this duct, then again for all a, b in sigma and(p, q)in q if we have this kind of transition delta (p, a) equal to (q, R); that means, is a right move over the head if the Turing machine in state p reading symbol a it will go to state q and the head be moved towards the right side by one cell, for this we will be using this two tiles it simply communicate head movement one square from left to right.

So, (p, a) is a content of this tile (p, a), and since head is moved towards right side this a will get toward it goes to the right side standing in state q, and it goes to right right side. So, the next case whatever it may be the symbol it may be b, so this we will be here, but the state will be this state q, and in both the cases we must have some tile below it. So, this communicates the movement of the head one square to the right, and the stands of state appropriately; similarly, for all a b and p q a b in sigma and p q in state q if delta p q equal to q n.

So, this type of type of tiles basically communicates the women of the head towards left side by one position. So, delta p a, so p a is the content of this tile, so a will be get proved we communicated upwards, so it changes state from p to q and it goes towards the left side. So, therefore, from left side it comes q whatever be the content earlier content of this tile. So, next it will be (q, b) the state will be changed to q and b will be carried forward and in both the cases we need to have some tiles below it, and then from the initial condition.

(Refer Slide Time: 24:31)

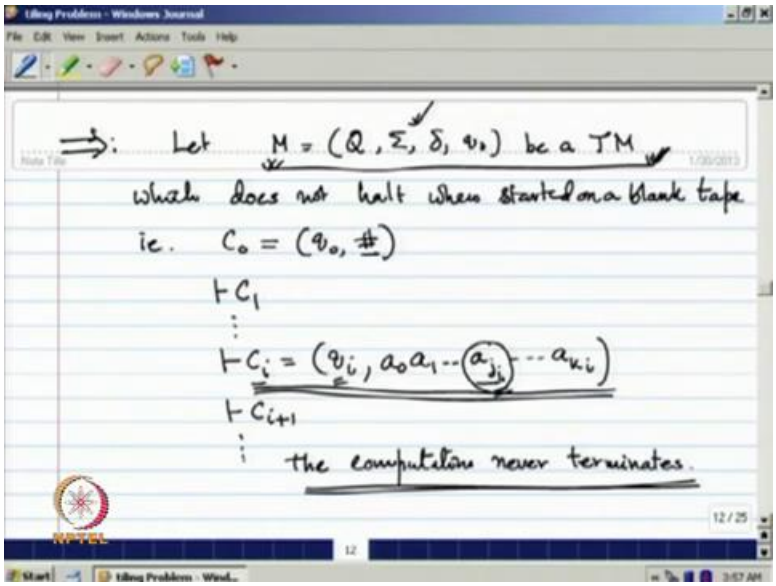


Since in the initial configuration we start with the state q 0 and blank, and towards the right we have only blank. So, therefore, we use this kind of tile to fill the for cell, and for all the remaining cells initially we using say blank and towards the left side and right side we have all blanks, and this blanks we carried forward, so that joined used this blank. So, in both these cases you just known that there is no necessary to have any other tiles below this tiles, so that is why we are not getting any doubts because those we used for initial configuration.

Now, we have constructed a tiling system, so basically in this tiling system of what you have done is that a soon what are the types of tiles to be used (Refer Time: 25:33), so all these types of tiles and we have construct those tiles from corresponding moves in the Turing machine, and the same time what I have done is that we said indicator here the vertical condition for example, here from a this a be carried forward. So, here again we will have (p, a) and it will be (q, b).

So, this gives the vertical condition and this case again we have this horizontal condition horizontal condition and so on. So, while constructing this tiles at the same time we have also indicated what should be the corresponding horizontal conditions, and vertical conditions by giving the content of the tile, and what should we have done top on the tile outwards right outwards left.

(Refer Slide Time: 26:39)

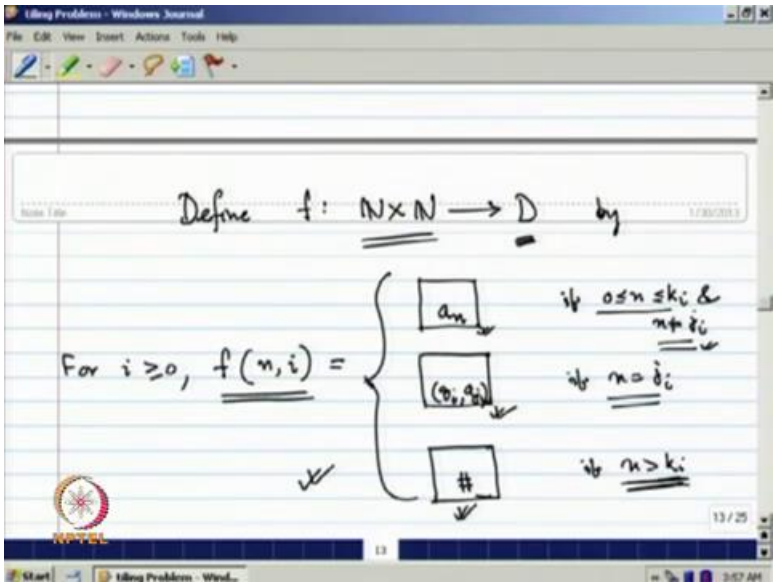


Now, when we show that given in Turing machine M is belongs to H 1 compliment if and only if D M has a tiling; that means, it is a reduction from the compliment of the compliment of this H 1 to the tiling probability given Turing machine M if it belongs to H 1 compliment if and only it belongs to H 1 compliment if and only if corresponding tiling system has a tiling.

Let us prove the forward direction suppose M is a Turing machine, with the following elements Q is the set of tile sigma is set of symbols delta is the transition, and q naught is the start state. And this does not halt when started on a blank tape because this is an instance of H 1 compliment since it is an instance of h one compliment this M does not halt once it is started on a blank tape. So, therefore, the first configuration will like this c 0, q 0, it starts from a blank tape therefore, the head is reading the blank and towards right side they are all blanks, then go to c 1 and c 2, and so on.

By computing according transition of the tile machine, and eventually suppose that particular configuration is c i will be of this form say q i a 0 a 1 say up to a d i and up to a k i. So, where a 0 a 1 up to a k i these are all symbols from the alphabet sigma, and then in going to next configuration c i plus 1 and so on and the completion never terminates they never incompletion it does not halt. Just consider this configuration c i considering this any arbitrary configuration c i, we will see the tiling we will give the tiling.

(Refer Slide Time: 29:02)



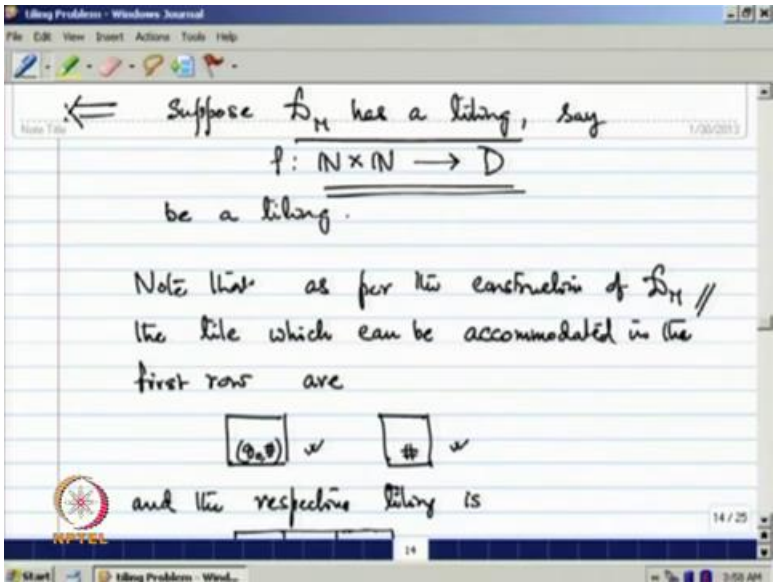
We give the tiling like this a tiling function, from N cross N to the set of tiles in the i d is the types of tiles we have already know how to construct from the tiles of from the construction that we have already described. So, for all i greater than 0 a pub (n, i) the what should the tile, for this quadrant (n, i) it will be a n simply a n the tile with content a n if n is greater than or equal to 0 and less than k i, but n not equal to j i. That means, it says that if it is within this other than is one it will not a j i it will retain the all the symbols that will be carried for.

So, a j will go to the next configuration a 1 will the next configuration up to a j i minus 1 will go to the next configuration in the upward direction. Similarly, a j i plus 1 up to a k i we will go upward in the next configuration in the next row, but if n equal to j i, then the constant tile will be q i a j i. So, because the constant is q i and it is kind of symbol say a j i, and if it is greater than q i, then after q i there is all blank because f come up to a k i. And, we started with an empty tape therefore, all the cells towards the right most symbol will be all blank.

And, hence if n greater then q i will be using this kind of tiles, so once we use this kind of tiles. So, corresponding to this termination we will construct the tiling system, and then if you consider the configurations start with c 0 we put it in the first row of the first quadrant on the acceptant, and then according to the mutual termination we will get the next configuration, and the tiles that we will be using be of the kind that we have already

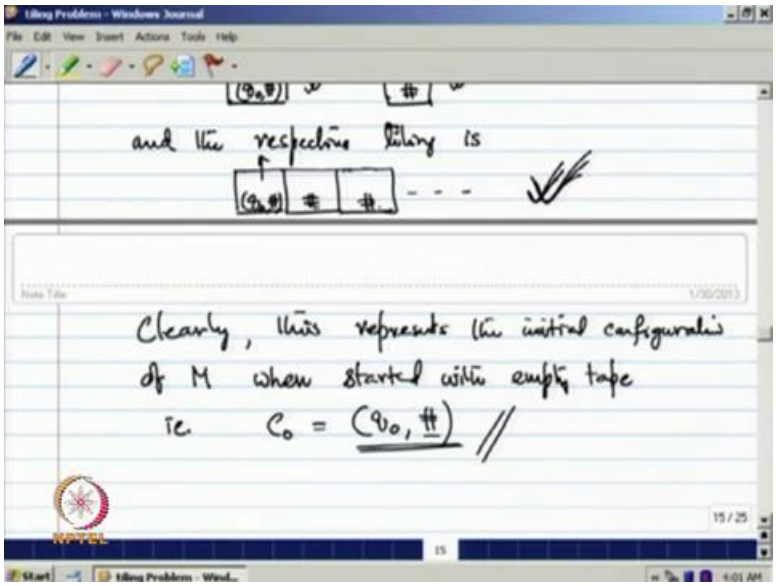
described according to the construction Turing machine and so on. And therefore, this gives a proper mapping work tile to be used for tiling the first quadrant. So, that way to be able to tile since the Turing machine never halts will be able to tile the first quadrant of the x y plane.

(Refer Slide Time: 32:37)



Now, consider a converse suppose the tiling system has a tiling, say there is a mapping from N cross N to D, so this is the tiling. Now, we know that that is for the construction of the tiling system D M from the Turing machines given Turing machine M the tile which can be accommodated in the first row horribly this kind q 0 blank and blank and there is no tile below it. Because, there is only possibility since with below it we do not have any other tiles.

(Refer Slide Time: 33:17)

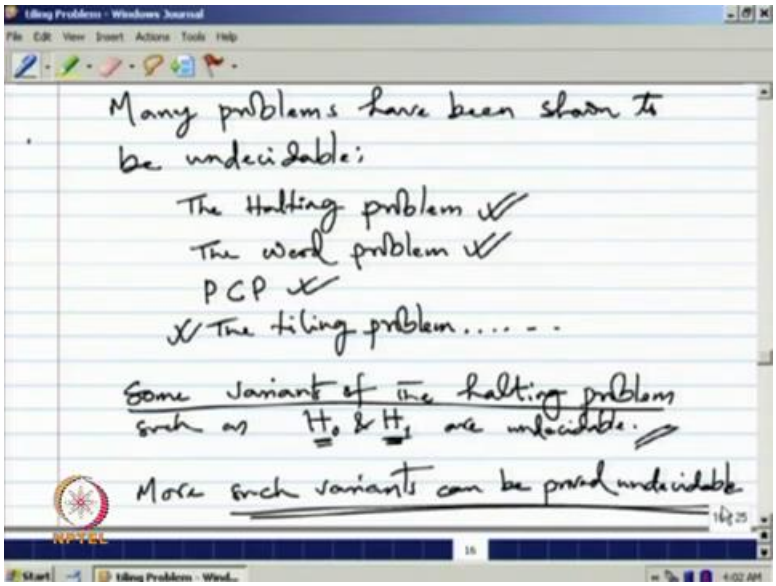


And, the respecting tiling must be q 0 blank then blank then blank and so on all blanks. Clearly, this represents the initial configuration of the Turing machine M when started with empty tape this must be in initial configuration. Therefore, c 0 is q 0 blank the head is reading the blank symbol, then since we have constructed the tiling system for the given Turing machine. So, what the next tile to be used above this will be defined or we will be decided according to the transition of the Turing machine.

Since, we have a tiling there must be some tiles to use out above it in the first row, and then in the next row, and this continue forever because there is a tiling that is what we have said because D M has a tiling. Since, it is a tiling will be able to fill up all the rows like this, but in this case what we have used is that this tiling system has constructed from the Turing machine only; that means, it clearly says that Turing machine has moves say tiling is you continuing infinitely. So, therefore, the computation of Turing machine also will go on infinitely; that means, from c 0 it will go to c 1 c 1 c 2 and so on from c i to c i plus 1 for all i.

So, it will continue forever so; that means, if there is a tiling there must be the Turing machine will never halt that is what we have got. So, therefore, so this shows that for any given Turing machine M if M there is a M belongs to H 1 compliment, if and only if D M has a tiling or D M is the corresponding tiling system.

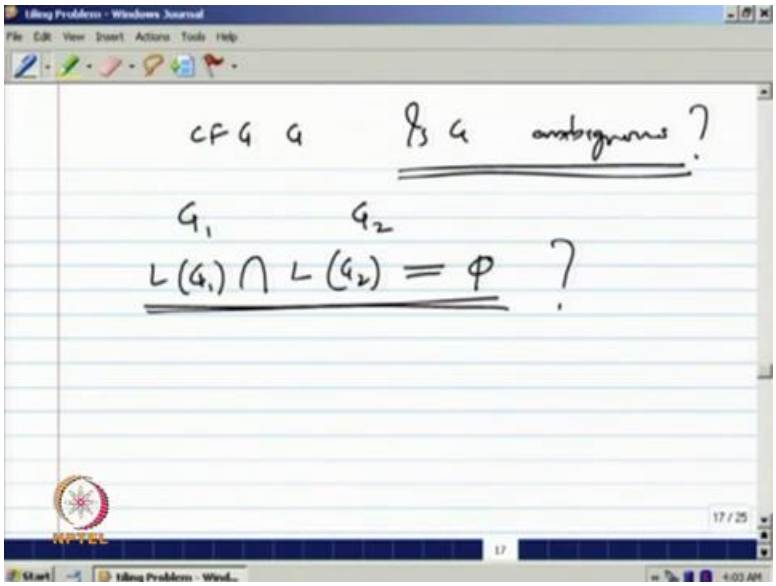
(Refer Slide Time: 35:55)



Now, we have already seen many problems that have been shown to be undecidable. For example, halting problem we have seen defined, and proved to what problem to undecidable we have simply defined P C P we did not proved of course, and now we have shown that defined the tiling problem, and shown it to be undecidable; similarly, you can keep on giving more and more problems to be undecidable. We have said that some variance of the halting problem for example, say H 0 and H 1 which are used in the context for example, H 1 is used to show that halting problem is undecidable sorry tiling problem is undecidable; similarly, H 0 was used to show that to what problem is undecidable.

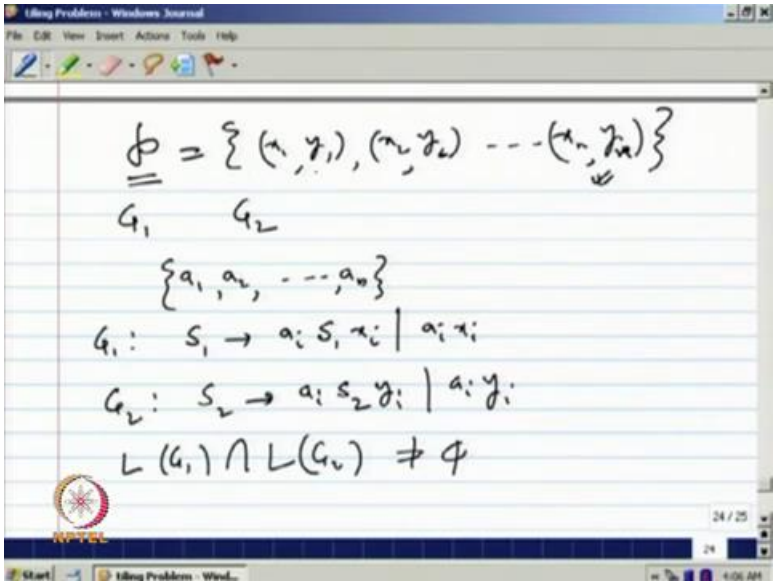
So, this variance of Turing halting problem H 0 and H 1 are also undecidable that we left as exercise more such variance of the halting problem can be defined and proved to be undecidable. Similarly, we can give we have shown that this to be a is undecidable, and then P C P can be used to prove that some problems let to say especially consider languages to be undecidable we have shown that given grammar.

(Refer Slide Time: 37:20)



Given a C F G where G is ambiguous is G because so this problem is basically, for any arbitrary C F G is undecidable. We used this P C P to show that the ambiguity problem of C F G is undecidable; similarly, we can show that given a C F G G L and C F G G 2 where L of G L intersection L of G 2 equal to phi. So, this problem is also undecidable, we can easily show it by using the same construction that was used in case of P C P in case of showing that ambiguity problem of C F G is undecidable.

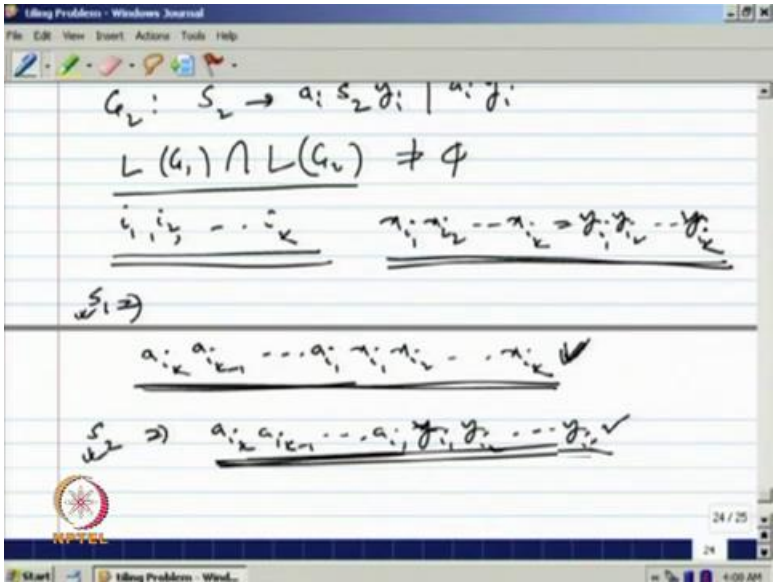
(Refer Slide Time: 38:48)



For example, just consider say there is an in set of P C P say it is x 1 y 1 x 2 y 2 up to say x n y n. So, from this in set of P C P we will construct grammars G 1 and G 2 we will construct conditional grammars. Such that, that P C P instances p has a solution if and only if L of G 1 and L of G 2 is non empty, now the construction likely similar that we have described in case of showing that a given adiabatic grammar is ambiguous not is undecidable.

For example, we introduce a new set of symbol say a 1 a 2 say up to a n as many symbols as they are listed as the pairs of pairs in the P C P, and then in G 1 we have productions like S 1 goes to a i S 1 x i and a i x i for all i similarly in G 2 we have S 2 goes to a i S 2 y i and a i y i. So, we can now easily show that L of G 1 intersection L of G 2 is non empty if and only if this P C P instance p has a solution. So, if because if this instance of P C P has a solution.

(Refer Slide Time: 40:52)

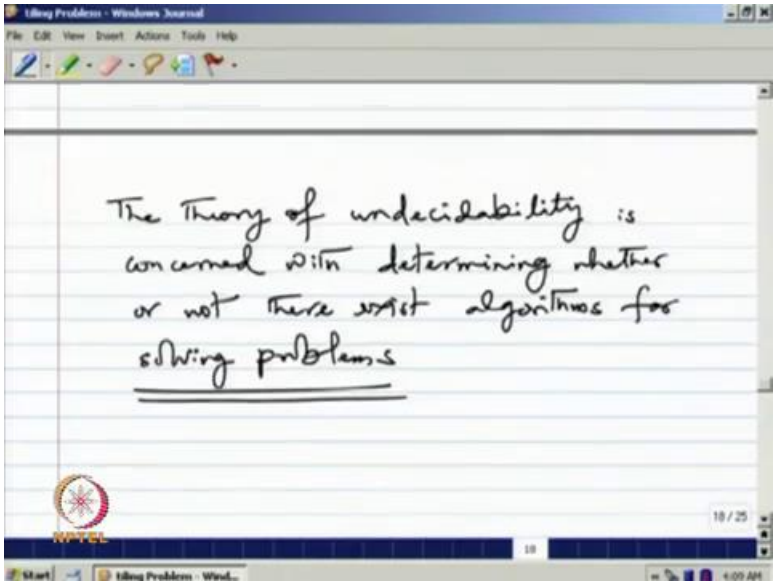


So, there must be some sequence say i 1 i 2, so i k which is solution to this P C P such that x i 1 x i 2 up to x i k equal to y i 1 y i 2 up to y i k. So, this term must be same because there is a solution to the P C P this kind of solution to the P C P, so according to the definition of P C P. So, this must be same, but in case of G 1 and G 2 we have seen that if you consider a i 1 or a i k a i k minus 1 up to a i 1 x i 1 x i 2 up to x i k. So, for this string there is a derivation in G 1 starting with S 1.

So, one derives this string similarly S 2 derives this string a i k a i k minus 1 up to a i 1 x, so in y i 1 y i 2 up to y i k. Now, since the first part is same a i k to a i 1 a i k to a i 1 and x i 1 to x i up to x i k this is identical to y i 1 to y i k, so this two strings only identical is the same . So, therefore, this string and this string belongs to both since this we started S 1 in G started S 2 in G some derivation we can although derived like this.

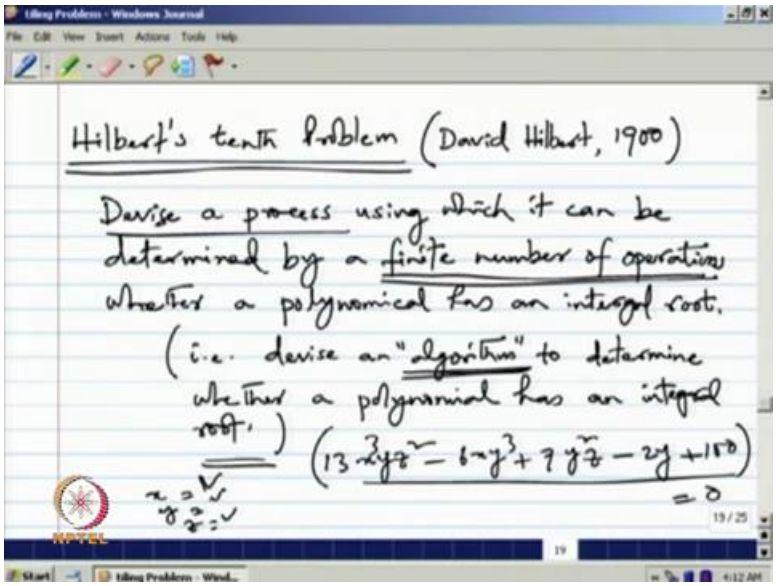
So, therefore, intersection L 1 and G 1 is not empty. So, this string is come on to both the languages L of G 1 and L of G 2 similarly, you can prove the converse; that means, the P C P solution if and only if L of G 1 and L of G 2 has common element. Now, similarly, we can define or give many other problems relate to constraint grammar and. So, that they are all undecidable by reducing P C P to those columns.

(Refer Slide Time: 43:41)



Now, please note that the theory of undecidable is concerned with determining whether or not there exist algorithms for solving problems. So, that is what we want to know does there exist an algorithm for solving any given problem if it exists, and this is decidable otherwise this is not decidable.

(Refer Slide Time: 44:19)

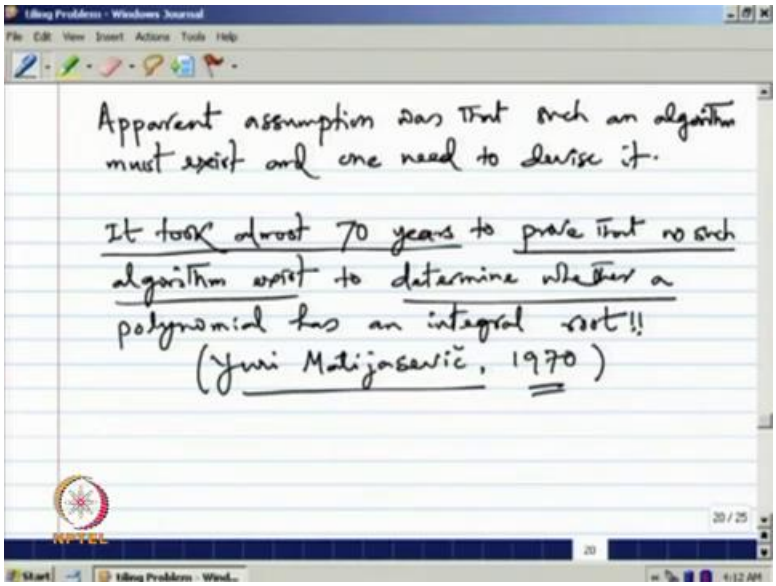


Now, people started viewing algorithms are most of the algorithms are known to people, since long is only the beginning of twentieth century 1900 that people started about thinking whether there exists an algorithm or not for any given problem. We started with an address given by David Hilbert in1900, so he passed many problems to the common area at the time.

So, one of those problems was like this is a tenth problem devise a process using which it can be determined by a finite number of operations whether a polynomial has an integral root; that means, devise a process devising a process using which determined by finite number of operations it means that he wanted to devise an algorithm that is that using an algorithm that is known to us devise an algorithm to determine whether the polynomial has an integral root or not that means if we have an polynomial like this say 13 x cube y z square minus 6 x y cube plus 7 say y square z minus 2 y plus 100 say it is a polynomial involving three variables x y and z.

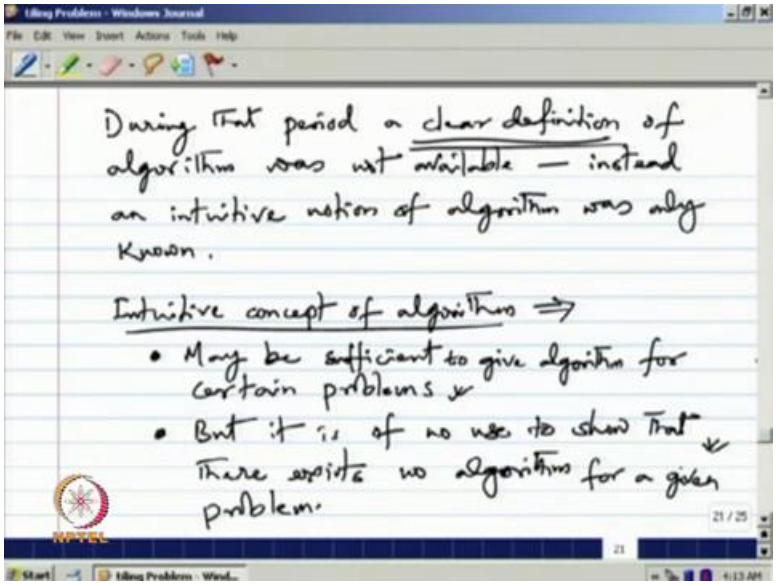
So, it will have root if we have an assigning some x equal to something some value of x y z such that those values when put here it satisfies zero it becomes equal to 0. So, it has a root. So, the question is whether we have an integral root whether some integers where we can put over here x y and z, so that it becomes equal to 0.

(Refer Slide Time: 46:48)



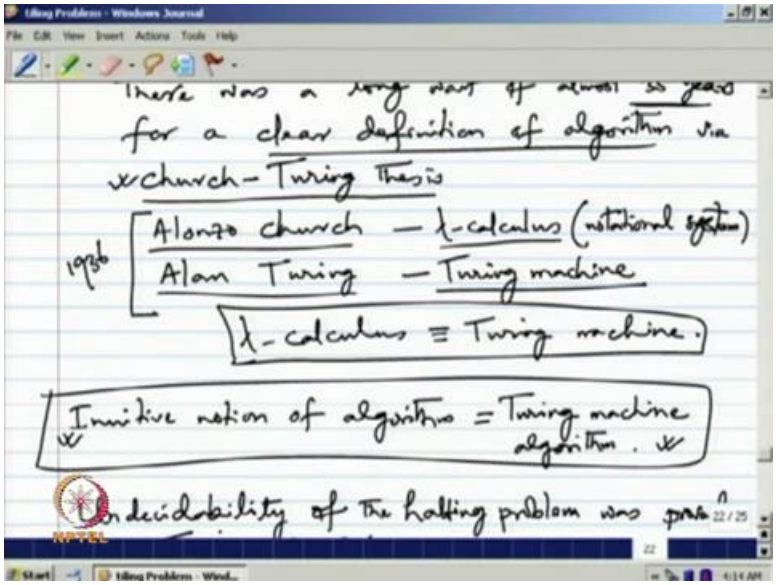
So, apparent assumption was that such an algorithm exist, so Hilbert assumption was that it exists and one need to simply devise this algorithm, but it took almost 70 years to prove that no such algorithm exists to determine whether a polynomial has an integral root this proved by matijasevic in 1970 it took long seventy years to show that no such algorithm actually, exists to determine whether any given arbitrary polynomial has a integral root or not.

(Refer Slide Time: 47:22)



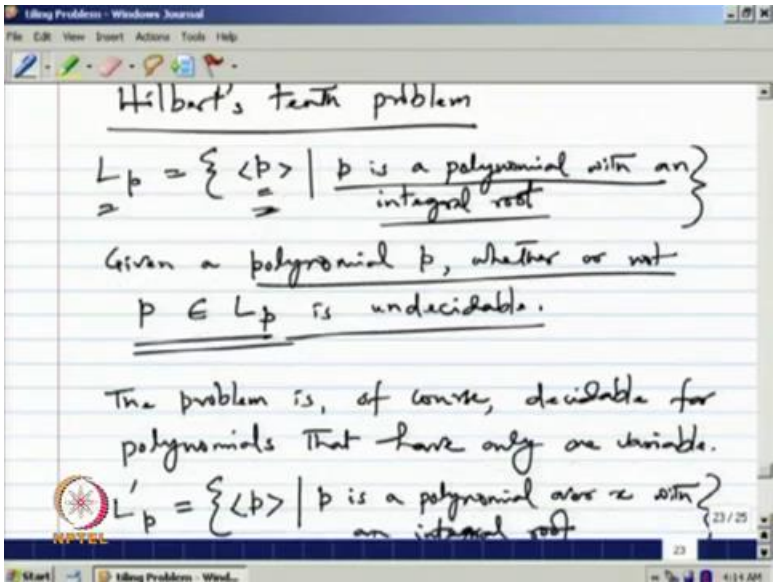
Now, during the period it cleared definition of algorithm was not available what is meant by an algorithm instead an intuitive most of algorithm also we known, but the intuitive concept of algorithm may be sufficient to give algorithm for starting problems, but it is not enough to show that there exist no algorithm for any given problem; that means, people are taking the time a clear definition of algorithm.

(Refer Slide Time: 47:55)



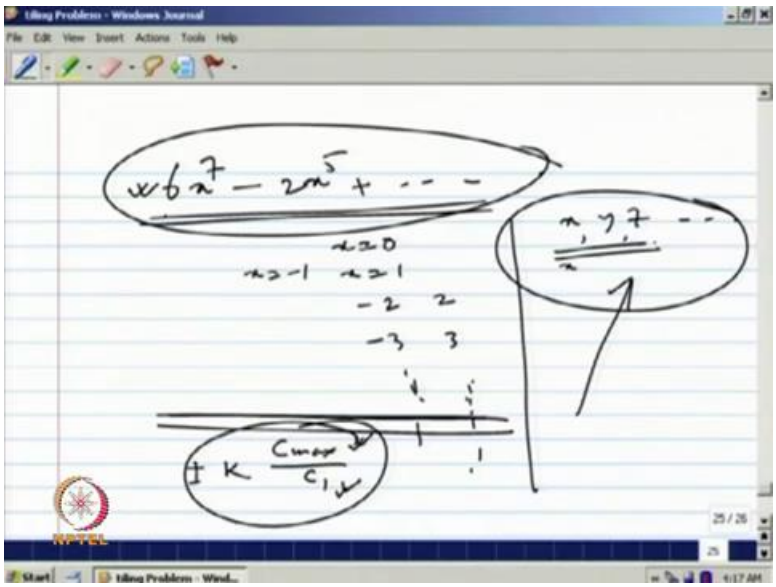
There is a long wait of almost thirty five years for a clear definition of algorithm an eventually given via such Turing thesis in 36 1936 Alonzo church developed lambda calculus using notational system. And Alan Turing developed Turing machine is computing or abstract machine and to show that lambda calculus and Turing machine this two notion are equivalent and intuitive notion of algorithm is equivalent to Turing machine algorithm this basically known as such Turing thesis. So, that means we can define algorithm by designing by giving a Turing machine and undecidable table halting problem was put by Turing in thirty six in thus Hilbert's problem

(Refer Slide Time: 48:50)



We can define the corresponding language like this say L of p is the set of all polynomial p is a polynomial with an integral root. So, the problem is given a polynomial p whether or not p belongs to L p is language is undecidable if problem is of course, decidable for polynomials then there are only one variable. For example, if you have, so variable involving on this x say

(Refer Slide Time: 49:32)



Suppose we have a variable i mean a polynomial say involve in 6 x to the power 7 minus 2 x to the power 5 plus and so on only x variable is only x. So, in such a case what a

Turing machine can do that it compute the polynomial for defined values of x starting at 0 put x equal to 0 compute the value of the polynomial if it is 0 its find will stop otherwise you compute the value polynomial for x equal to 1, and x equal to minus 1 if at any point it will becomes 0, then stop otherwise you continue in both directions which 2 and minus 2 then minus 3 and 3, so it will continue and so on.

Until it finds a solution, but point is that such same thing can be used for multi value multi variable problem also for x y z and so on. Suppose we have take different values of x y z and keep on finding the polynomial until it becomes 0; that means, we have Turing machine to do this, but whether it is a decidable or not that is not known we have single variable we need to go infinitely many steps. So, we can stop at some point when we do not want infinite many step, and see that this has no integral root or if you finally, will stop the Turing machine stop saying that it has an integral root, so up to what limit.

It will go it can be said that it is plus minus k some c makes verses c 1 where k is the number of terms in the polynomial how many terms are here, and c makes is the coefficient with largest absolute value and c one is basically the coefficient of the highest order term. So, you will keep and going like this until up to this limit until it finds a root if it does not finds a root we will say that declare that there is no integral root for this polynomial. So, therefore, for single variable the problem is decidable and for multiple variable the problem is not decidable, and it has proved by martijisavic in 1970 only that for multi variable case the problem is undecidable.