Theory of Computation

Prof. Kamala Krithivasan

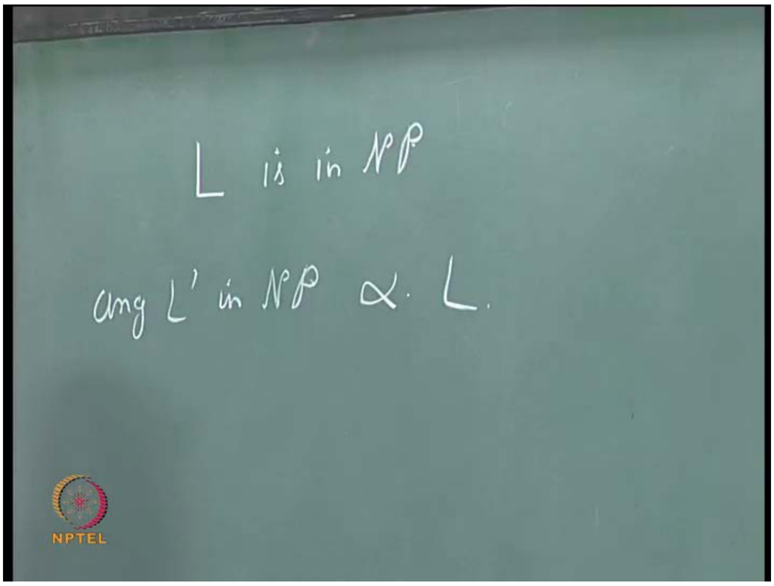
Department of Computer Science and Engineering

Indian Institute of Technology, Madras

Lecture No. # 37

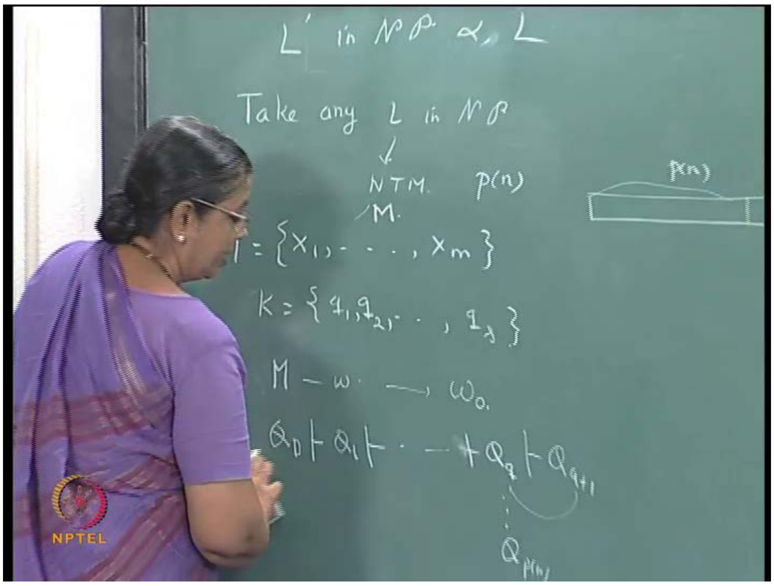
NP-Complete Problems (Contd.)

(Refer Slide Time: 00:26)



So, we were considering the problemof Booleansatisfiability.We are trying to show that Booleansatisfiability is NP-complete.How do you show that a problem is NP-complete,when do you say a problem is NP-complete or a languages in NP-complete?If L is in NP any L dash in NP is polynomially transformable to L these two conditions have to be satisfied.Now, we have seen that given a Boolean expression non-deterministically,you can guess assignmentandevaluate that in order n squaredtime.If the length of the Boolean expression is na non-deterministically youcan evaluate in order n squaredtime.

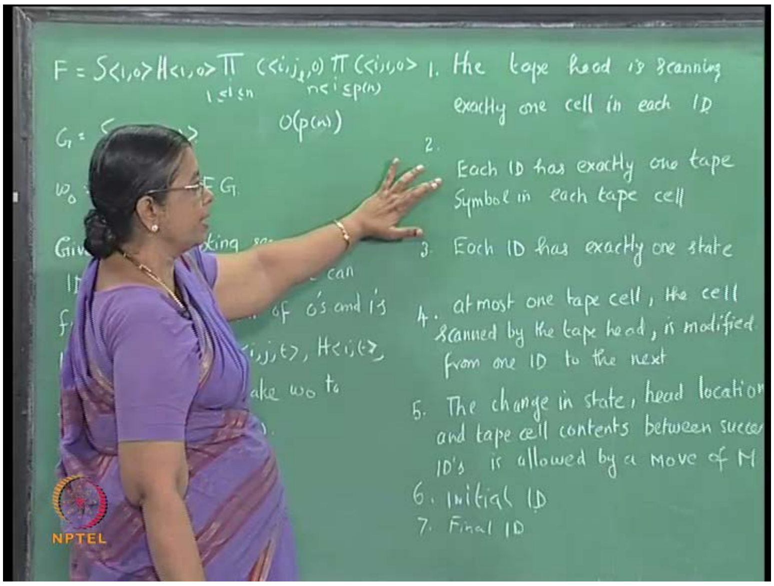
(Refer Slide Time: 01:02)



So, the otherthing you have to prove this any L dashin NP is polynomially transformable to L or Boolean satisfiabilityfor that.Takeany L in NP take any language L inNP. L is accepted by a non-deterministic,Turing machinein polynomial timep n and it has gotsymbols X 1, X 2, X m this is thetape alphabetand the state alphabetset of statesis given by q 1, q 2,q s.Now, q 1 is taken as a initial state and q 2 is taken as a final state X1,X 2, X m are the symbols, this is the blank symbol, this is taken as the blank symbol.

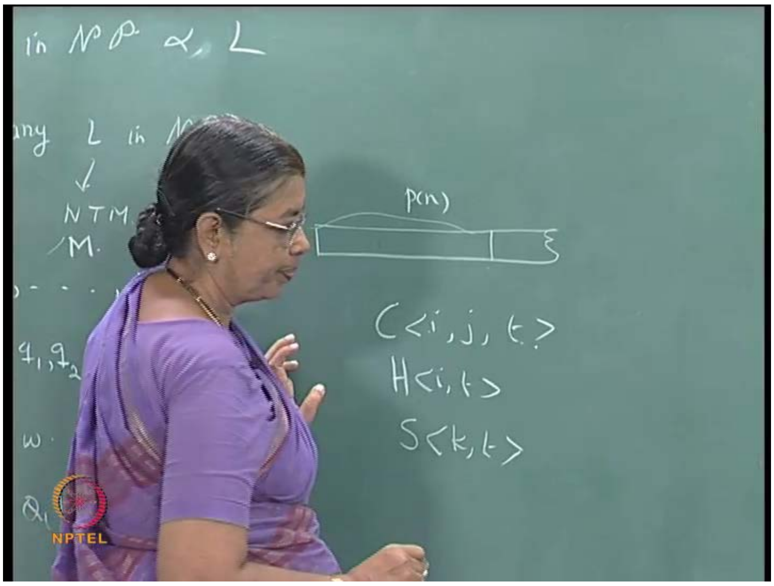
Now, if theword given M wgiven M.This is the Mgiven M and wyou can writeBoolean expression w 0.Now, given M and w there will besequence of ID's Q 0,Q 1,Q 2 up to Q qwhere it will accept Q will be less than or equal to p n, but Q qplus 1 you take to be the identical with Qq same afterwards up toQ p n,it is sameand the tape cells.It uses at most p nand without loss of generality, you assume that every ID has p n cells.So, with this assumptionsyou write down the expression w 0 and how do you go about writing w 0.

(Refer Slide Time: 03:09)



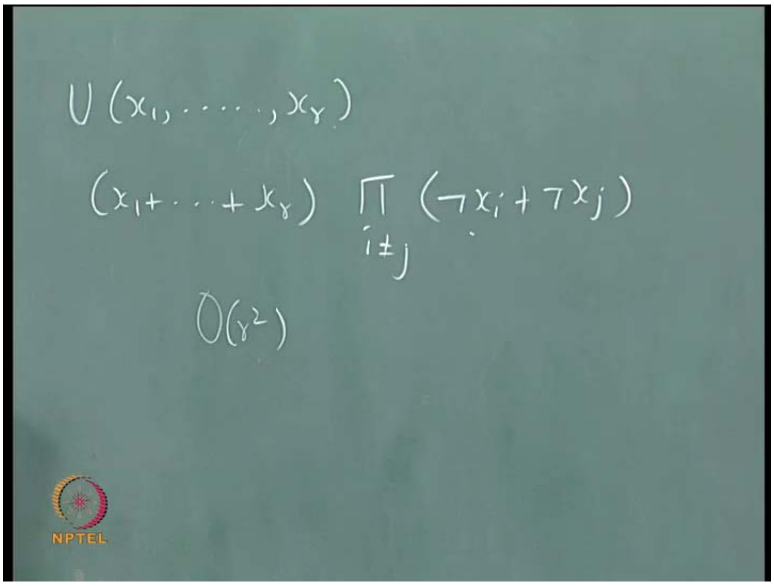
This we have consider in the last lecture itself, but let us repeat you have these conditions.The tape head is scanning exactly 1 cell in ID for eachID has exactly 1 tape symbol, in each tape cell and each ID has exactly 1 state at most 1 tape cell.The cell scan by the tape head is modified from 1 ID to the next ID and the change in the state, the head location the tape cell contents between successive ID’s is allowed by a move of the Turing machine, then you have to specify the initial ID and the final ID for these 7 conditions you write expressions.

(Refer Slide Time: 03:57)



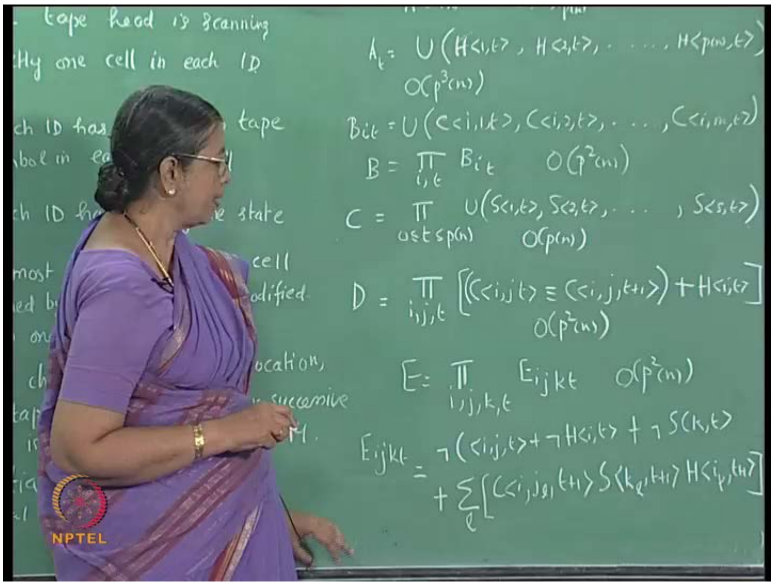
Boolean expressionsmaking use of Boolean variablesCi, j, t.Whatis the meaning of that the ith cell contains the jth symbol at time t.H i, tmeans the head is scanning the ith cell at time tand S k, tmeansthat state at time t is q, k.So, these Boolean variables you define.

(Refer Slide Time: 04:28)



Now, we make use of 1 expression Ux 1, x 2,x rand this is defined as x 1 plus x 2plus x r,phi,i naught equal to jnaught of x iplus naught of x j.This the length of this expression is order r squaredand this is equal to 1 exactly.When 1 of this 1 others are0this all we have seen.Now, how do you write the expressionsfor them the tape head is scanning exactly 1 cell in each ID.

(Refer Slide Time: 05:08)



So, you write at time t this is the expression H 1 t,H 2 t,H p n, t only 1 of them will betrue rest of them will be0.And you have to specify that from the zeroth instance top nth instance.So, the length this itself is of order p squared nand you have p n of them or p n plus 11 of them.So, the length of this expression will be order p cubed n.The second condition is each ID has exactly 1 tape symbol in eachtape cell.So, you write it as expression B.

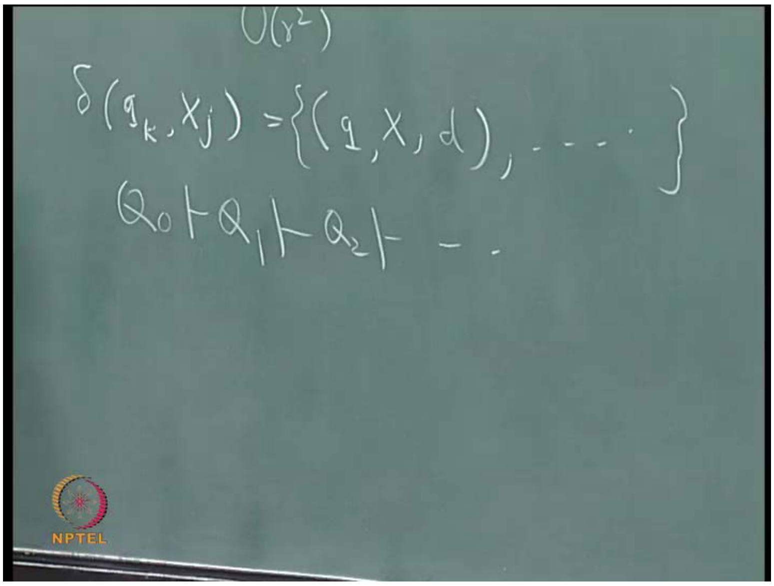
B is product of B i t where it ranges over i and t and B i t is the ith cell at time t contains either symbol 1 or symboltwo or symbol three or symbol n only one of them will be truethe ith cell at time t can contain only one of them.

So, this is there are m symbols here so the length of that is order m squared, but m is a constant m and s are constant.So, that is constant, but this itself ranges over i and ti ranges from 1to p n t ranges from 0 to p n.So, it is order p squared nsimilarly,the expression for this thing is each ID has exactly 1 state.So, the state can be only one of them first state 1 or state 2or state S only.One of them will be true that is given by the, this expression and you have to take for all t this is for 1 particular t you have to take for all t the length of this expression will be orderSsquared, but S is a constant.So, the length of this will be order p, n.

Then, the next condition is at most 1 tape cell.The cells can by the tape head is modified from 1 ID to next ID.So, the expression for that is D the product of some expressions like this, where each product tells that either the head isscanning the ith cell at time tin that case, the cell the contents will change otherwise thei th cell if it contents the jthsymbol at time t.It will continue to have the same symbol at time t plus 1,theidentically equal to here again,this the length of this is fixed, but you are taking the product to overi, j, t.J will vary from 1 to m which is a constant, but i and t will vary from 1to p n and 0 to p n respectively.So, its order p squared n.

Then the fifth condition is the change in the state head location and tape cell contentsbetween successive ID’s is allowed by a move of m.

(Refer Slide Time: 08:32)



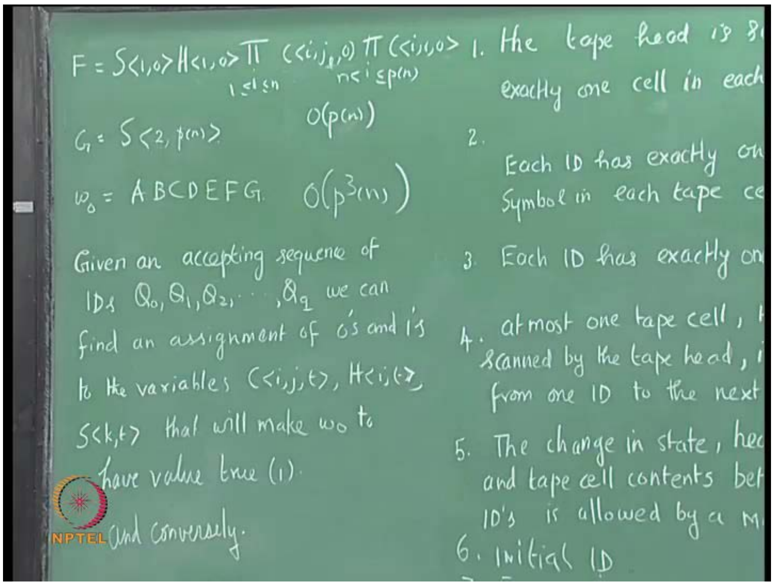
Now, you must remember that we are considering a non-deterministic Turing machineand if you have a, q, k delta of q k,X jthere will be one possibility will be q,X, dthere will be several possibilities like that,finite number of choices.So, what we are considering when I speak of ID q 0,Q 1,Q 2 this isonesequence, we are considering.One sequence which is leading to acceptance there may be many sequences one of them we are considering.

One of them which is leading tothe acceptance, we are considering that you must rememberand the fifth one is the change in state head location and tape cell contents between successive ID’s is allowed by a move of the Turing machine.The expression for that is E.E is the product of expressionsE i, j, k, t, but what is E i, j, k, teither the ithcell does not content,the symbol j or the head is not scanning the ith cell or the state is not q, k at time t.

So, if the state is q, k andhead is scanning the ith cell and the ith cell is containing X j then the next move has to be specified like this then maybe finite number of choices.1 of them if you take q,X, d that is given by this.The ith cell will contain the symbols specified by jl,jlis X.If we look into that at time t plus 1, the state will change from k to klsome other state and the head position will change to i minus 1 or i plus 1depending upon whether, the move is left orrightyou are having a finite number of choices, of moves.

that is why this sigma any one of them will be true, you choose one of them.So,one of them will be truethen you have to specify the initial ID and the final ID before thatthe length of this expression each 1 is of finite length, but it is a product over i, j, k, t.J ranges from 1 to m, k ranges from 1 to s.So, they are all constant, but i ranges from1 to p, n and t ranges from 0 to p, n.So, the length of the expression will be order p squared n.

(Refer Slide Time: 11:31)



Now, the initial ID is specified like this you are taking t is equal to 0 and initial state is q 1.Head is pointing to cell 11 initially 1,1 0.And the first n cells contain some symbols the input right rest of them contain, the blank from n plus 1to p n, it contains the blank symbol first n cells contain theinput,the finalthing is at time p n, q 2 is the state q 2 is taken as a final state.Now, if even it goes to a final stateat an earlierinstant the remaining ID’s you keep them as the last Qq is the final ID say it stops.

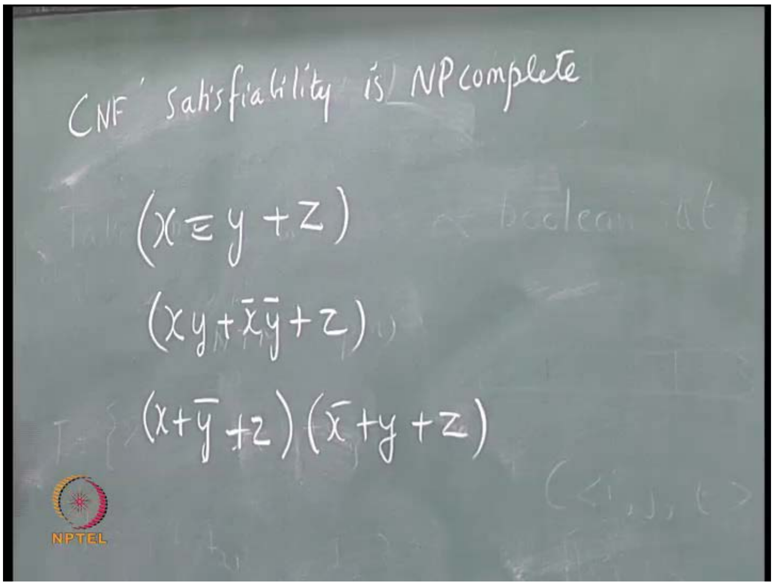
At that point then Qq plus 1, q plus 2 we take as identical toQq.So, anyway at time p, n it will be in final state so, the expression w 0 is given as the product A, B, C, D, E, F, Gand what can you say about this length of this each 1 is at most order p cubedn, psquaredn, p,nor something like at most p cubed nand there are several such expressions.So, at the most it will be order p cubed nthe length of w 0 is orderp cubed nand you can write it down in a time proportional to that.So, given m and w we can write down w 0in a time which is polynomial that is where the polynomial transformability comes.

Now, we can very easily see thatthe way we have find the Boolean variablesC,i, j, k, t,H,i, t,S, k, t etc.The way we have defined them we can see that if there is an accepting sequence of ID’sQ 1,Q 2,Qqthen you can find an assignmentto the variablessuch that, w 0 will evaluate to 1 or 2.Conversely, if there is an assignment which evaluatesfor which w 0 evaluates to 1 that means.Successive ID’s you can havewhich leads you to acceptance.So, this showsthatany L is polynomially transformable to theBooleansatisfiability problem.So,Booleansatisfiability is np-complete.

(Refer Slide Time: 14:51)

Now, once you have shown that Booleansatisfiability isNP-complete.1 problem you know that is NP-completethis is the clause, NP and NP I keptthis is the clause p whether, they areequal isnot known this is NP complete.Now, if I know there is one language L 0, which is Boolean satisfiability, any other problem if I have I can show that.This is can be transformed to that and if this can be polynomially transform to another problem, that will be also NP-complete provided it is in NP.So, that is what you make use offor proving the clique problemvertex cover problem and so, many other problems you reducethe3 sat problemor Booleansatisfiability.

(Refer Slide Time: 15:54)



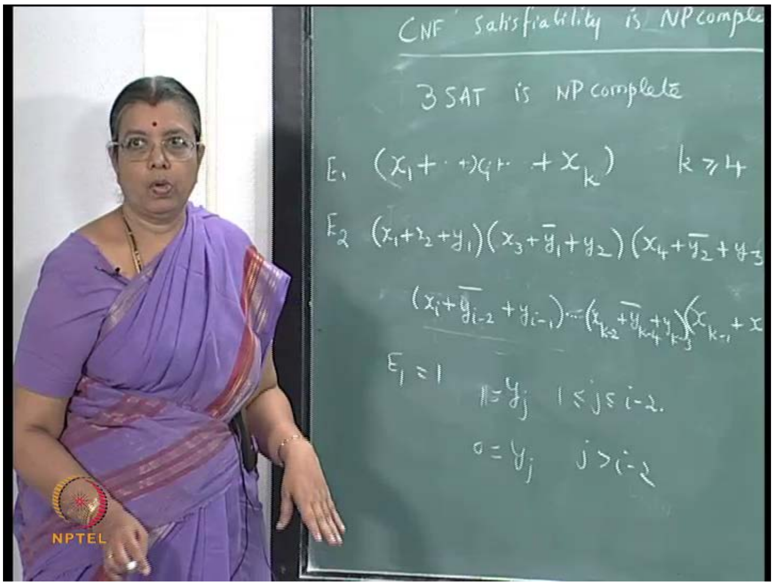
Now,generally you take theexpression in CNF.CNFsatisfiabilityis NP-complete.So, here we are making 1 more restriction that the given Boolean expression is in CNF.The same proof we are going to follow same proof we will holdonly 1 or two places it is not in CNF rest of the thing it is CNF.The way we have written the U function it is in CNF and this is in CNF everything is in CNF except D and E.Except D and Erest of them are in CNF.A, B, C are making use of thatU. That Uexpression itself is in CNF and the last two expressions F and Gtrivially, they are in CNF they are product of single literals they are also in CNF.

So, the only problem for this will be D and E.Now,E itself if you take it is a product of some expressions like this and these expressions are of finite length.They are of finite length using r 0 etc.And any finite length expression you can bring into CNF,you know that any expression of finite length, we can bring it to CNF.So, each 1 of Ei, j, k, t you can bring to CNF by a procedure and that should not take longtimebringing to CNF again, will not take much of time and the length also will not increase toomuchit may increase by a constant factor.

So, this is the product of such expressions so, it will be you can bring it to CNF and the time taken the polynomial or non-polynomial not will is not going to get affected.D is of the form something equivalent to something plus something, this is fixed length, but again this is the product over i, j, k, t.So, it is of this form if you take D it is of this formx is identically equal to yplus z.Each expression is of that formthis you can write as x, yplusx bar, y barplus zor this is equivalent to x plusy barplus z.x bar plus yplus z.This will be equivalent to thisconvinced x bar and C this is equivalent to this and this is in CNF right this is in CNF.

So, each factorthe expression D you can bring it to CNF.So, everything you can bring to CNF and the time taken to do thatis not too much.The length when you bring it to CNF,the length will not increase too muchisn't it by linear factor or by a constant factor only it will increase.So, the length of w 0 is still again a polynomial in L,we had p cubed n may be the constant factor will get slightly affectedyou will get an expression.So, the same proof with slight modification you can usefor showing that CNF satisfiability is NP-complete.

(Refer Slide Time: 20:37)



Now, we are assuming that the given Boolean expression in CNF.rightThen 3SATisNP-complete,we put some more restriction that the expressions have to bein conjunctivenormal form that is it is a conjunction of disjunctions, but each one has onlythree literalslike thisbar each factor has only three literals, that is variables or negation of the variables.How do we prove this?Any expression say x 1 plus x 2plus x k.This is in this is the factor of theCNF,k isgreater than or equal to 4.

Now,I want to find an expressionwhich is equivalent to this such that, if this is satisfiable the other one is satisfiable and vice versa.And the expression which I am going to write is in such a way that each factor has only three literals, it is in three sat form this is 1 expression E1.I am going to write an equivalent expression E2,Iwould not say equivalent where we make another expression E 2,where we make use of more variables.Now, if this is satisfiable if there is an assignment which evaluates this 1 to 1 there will be another assignment which will evaluate E 2 to 1 and vice versa.

How do I write the expression E2?You introduce variables this is x k. x 1, x 2,x k.So, you introduce new variablesy 1, y 2,yk minus 3.New variables Boolean variables you introduced and write an expression like thisx 1 plus x 2plus y 1x 3 plusy 1 barplus y 2,x 4plus y 2 barplus y 3and so on.General term will bex iplusy i minus 2 barplus y i minus 1.And you proceed up tolast 1 will bex k minus 1 plusx kplus y k minus 3.The previous 1 will bex k minus 2,x k minus 2 plus yk minus 4 barplus y k minus 3and so on.

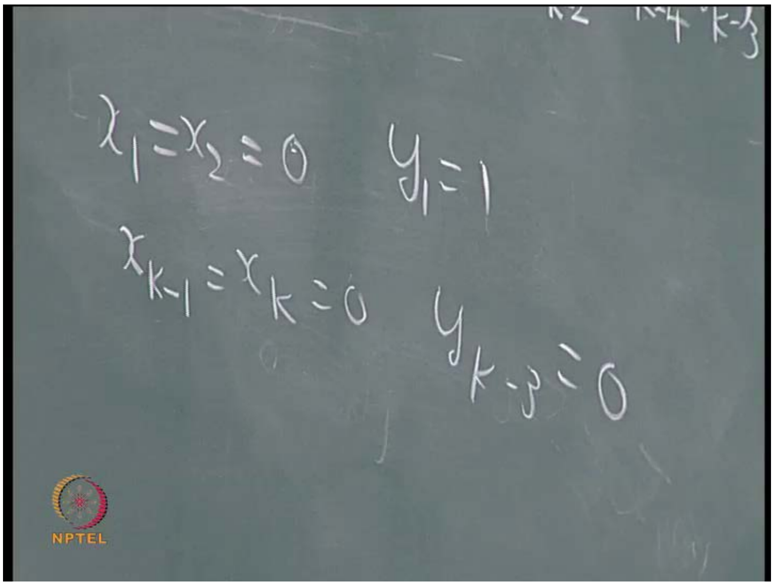
First factor you introduced y1,the second factor the negation of that and y 2 then next factor negation of y 2 and next one and so on.Until for x k minus 2, the negation of y k minus 4 and y k minus 3then the last factor is x k minus 1 x k and negation of y k minus 3.Now, if there is an assignment to the variables which makes this equal to1there is an assignment to the variables, which makes this expression equal to 1and if there is an assignment, which makes this expression equal to 1 there is an assignment, which make this expression equal to 1how do you prove that.

Now, supposeE 1 is equal to 1, that is E1 evaluates to 1 that is, there is some x iwhich is true.Some x i must be 1 then only that expression will evaluate to1and look at this factorhere it is 1.So, all y j where j is between 1 and i minus 2,make them equal to 1give the assignment value 1 give them value 1and all y jsuch that, j is greater than i minus 2 make them 0.

So, what happens y 1 is 1.So, this will be 1, y 2 is 1.So, this will be 1 each of the factors will be 1and look at these factorswhen it is greater than i minus 2 it is 0.So, the barred version will be true this is 1, this is 1 and so on.So, those factors will be true, but look at this factor this is 1 so, barred version this is 0.Y i minus 2 is 1, y i minus 2 baris 0,y i minus 1 is 0,but x i is 1.

So, this factor will also evaluate to 1 so, each of the factors evaluates to 1.So,if there is an assignment which makes this expression equal to 1there is another assignment which makes this assignment equal to 1.

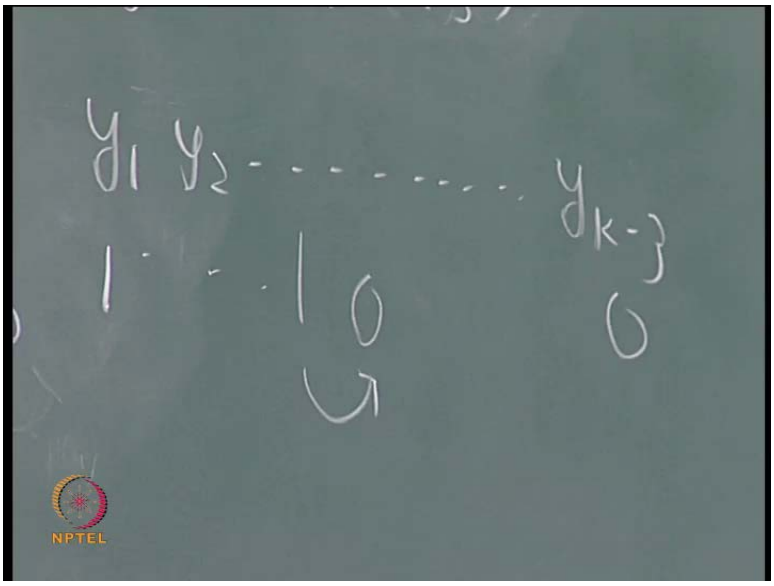
(Refer Slide Time: 27:31)



And converselyE 2 is equal to 1 implies E 1 is equal to 1.I mean E 2 is equal to 1 what I mean is E 2evaluates to 1 then,E 1 evaluates to 1.How do I prove thatthis expression evaluates to 1?So, each 1 of the factor should be 1,if each 1 of the factor is 1, then I say that this also evaluates to 1you start with that suppose x 1 or x 2 is 1 obviously, the result holds.If x 1 is 1 or x 2 is 1thisx 2 is 1 this will evaluate to 1.

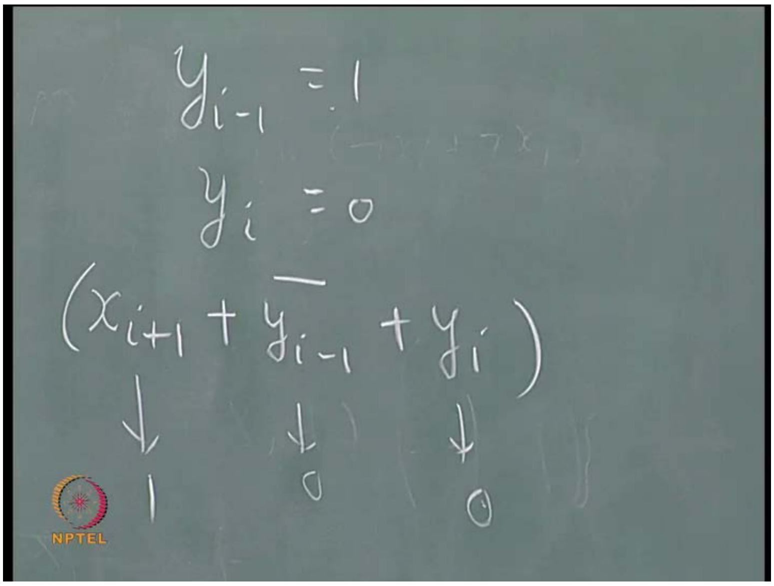
So, you have to consider the casethe x 1 is x 2 is equal to 0 and similarly, if x k minus 1 is 1 or x k minus 1is 1 or x k is 1 then,1 of them is 1 means, this will also be 1.So, you have to consider the case when x k minus 1 is equal to x k is equal to 0.Now, this factor must evaluate to 1, if x 1 is equal to x 2 is equal to 0,y1 must be 1and if this evaluates to 1if this is 0,this is 0,this must be1 that is y k minus 3 must be equal to 0,then only the barred version then a negative version will be 1.

(Refer Slide Time: 29:19)



So, if you look at the assignment for y 1, y 2,y k minus 3this is 1 this is 0 so,somewhere it will change from 1 to 0 at some positionit will change from 1 to 0 it may change in many places but, atleast 1 position.

(Refer Slide Time: 29:49)

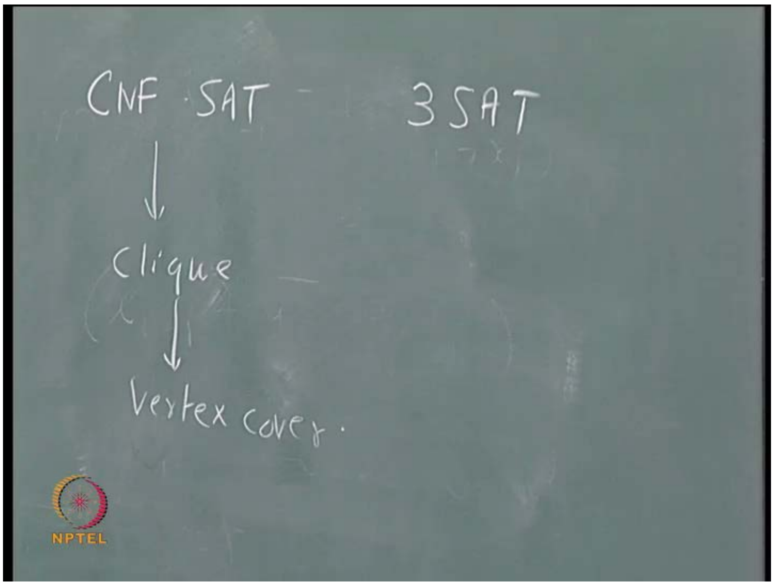


It will change from 1 to 0 that isy i minus 1 is 1,y i is 0.And herewe have a factorx i plus1 plusy i minus 1 bar plus y ithere will be a factor like this.And this is y i minus 1 is 1 though this is 0 this is 0.So, this has to be 1 every factor must evaluate to 1.So, there should be 1 variable which evaluates to 1 so, if there is 1 variable whichliteral which evaluates to 1 then this expression becomes 1.so, if this is satisfiable,this is satisfiable and if this is satisfiable,this is satisfiable.

What can you say about the length of this expression?It will be some constant times the length of this some seven or eight whatever it is, but each 1you have a factor whose length will be some 1, 2, 3, 4, 5, 6, 7 or something like that.So, the length is only increased by in linear way I mean 7 times.If the length of the original expression is n this will be some 8 times n or something like that.So, from this you can write down this in a verysystematic manner and that is not going to take a lotgoing to take linear amount of time.

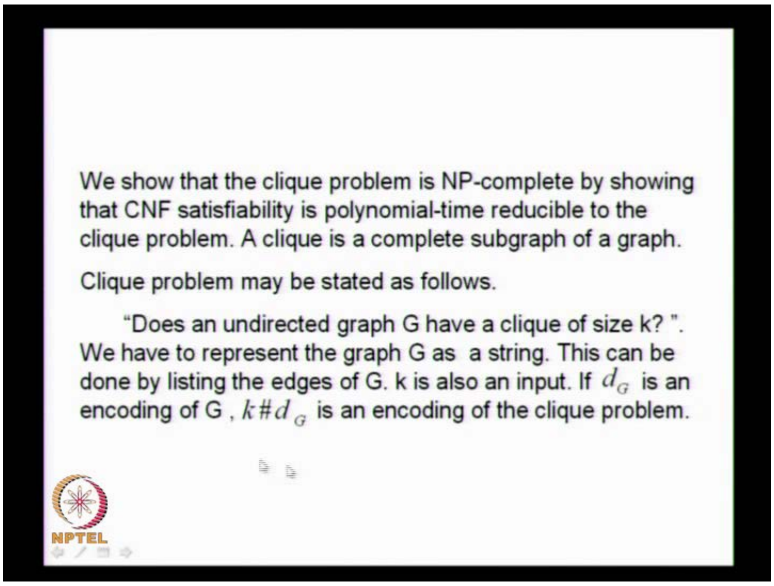
So, in the original problem you had everything in CNF the first 1,some expressions were not in CNF,then we saw that they can be converted into CNF.Now,once the expression is CNF,each 1 of the factors we can make it intoexpression like this where each factor has only three literalsand the time to do that will be only linear in terms of the length of the original expression.This is the polynomial or the non-polynomial this I mean the polynomial question is not affected.

(Refer Slide Time: 32:59)



So, from this we conclude that3 sat isbecome linear.So, once you havethis 3SATyou have CNF satisfiability,3 sat many problems we can immediately, reduce this you can reduce to clique,this you can reduce to vertex coverand so onlike that.Once, you have 1 problem which is known to NP-complete, you can reduce it to some other problem by a polynomial transformation and so that, that is NP-complete this is the method which isfollowed.

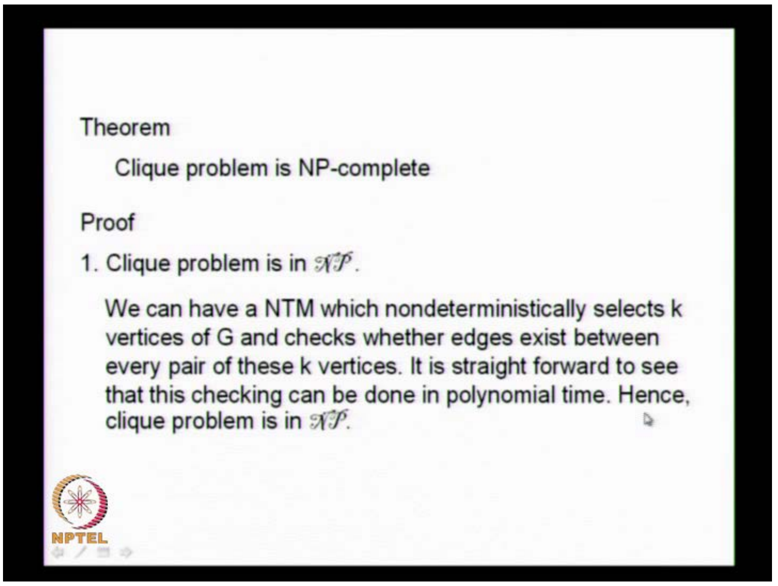
(Refer Slide Time: 33:51)



We shall show that the clique problem is NP-complete by showing that.The CNFsatisfiability is polynomial-time reducibleto the clique problemactually, once we know that a problem is NP-complete,we can reduce it to another problem and show that the new problem is NP-complete that is what, we are going to do now.We know that CNF satisfiability is NP-complete.Now, we are going to reduce it to the clique problem now,

What is the clique problem?A clique is a complete subgraph of a graph, the Clique problem may be stated like this.Does an undirected graph G have a clique of size k, we have to represent the graph G as a string.This can be done by listing the edges of G.k is also an input where you have to find out, whether it has a clique of size k.If d G is encoding of G then k hash d G,you can take it as the encoding of the clique problem.

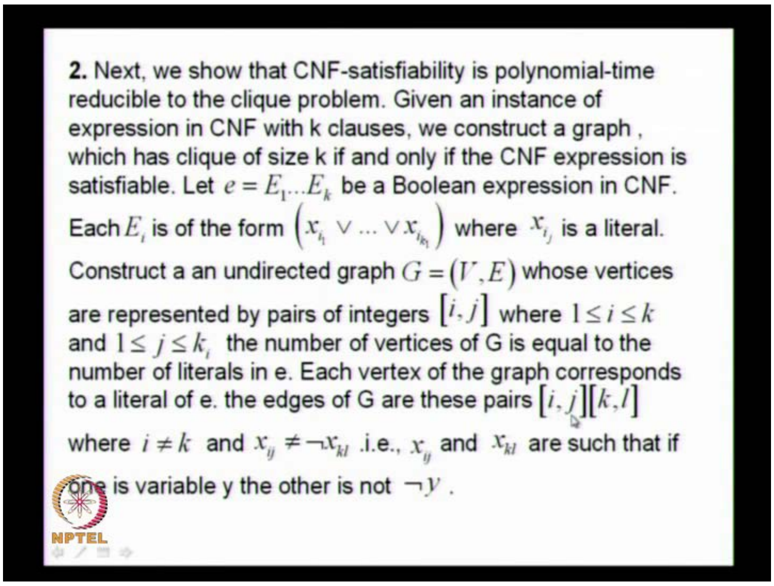
(Refer Slide Time: 35:00)



Now, we want to show that the clique problem is NP-complete.To show a problem is NP-complete two things you have to do.1 is you have to show that it is in NP and then we have to reduce an known NP-complete problem to that.So, to show that it is in NP what we do is.We can construct a non-deterministic Turing machine which will accept this, in polynomial time.So, we can have a non-deterministic Turing machine, which non-deterministically it selects k vertices of G and then it checks whether edges exist between, every pair of these k vertices.

If edges exist between every pair of these k vertices then, there will be a subgraph which is a complete subgraph of size k that is a clique.So,nondeterministically, it can select k verticesand do this it is straightforward to see that this checking can be done in polynomial time.So, we have proved that the clique problem is in NP.

(Refer Slide Time: 36:12)



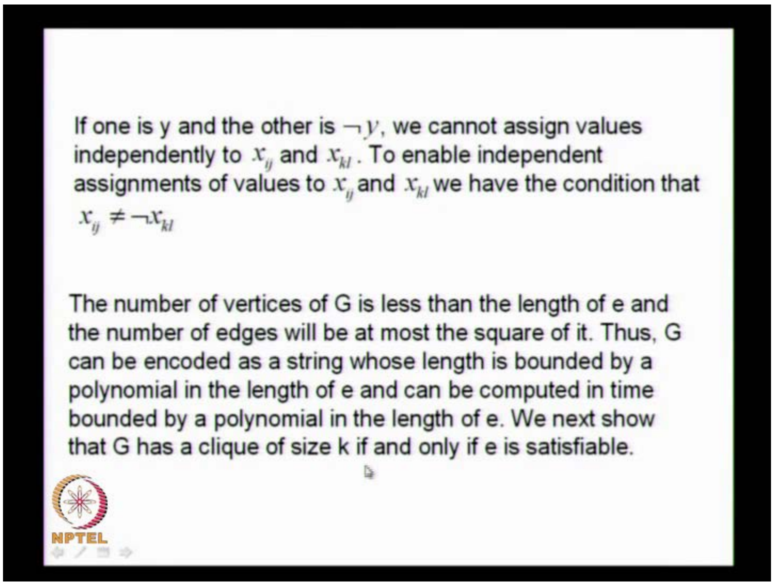
Now, we have to show that the CNF satisfiability is polynomial-time reducible to the clique problemgiven an instance of the expression in CNF with k clauses, we construct a graph which has clique of size k, if and only if the CNF expression is satisfiable.Let us take the CNF expression is e that is E 1, E 2, E k there are k clauses each E i clause this is the Boolean expression.Now, each E i is a clause and it is of the form X i 1 or X i 2 or etcetera Xi k 1there are k 1 literals in this clause.

So, each X i j is a literal it is a variable or the negation of the variable.Now, construct an undirected graph in the following manner given an instance of c s. CNF you construct a graph in the following manner G has vertices V and edges E,the vertices are represented by pairs of integeri, j.Where I will be from 1 to k and j will be from 1 to k ithat isthe first component of the label of a vertex will be 1 to k that is k, clauses it will denote the clause which the variable will belong.

And j will denote the literal in that clausesuppose, you have i, j that means it is corresponding to theith clause and the jth literal in that clause.So, j will vary from 1 to k,i.The numberof vertices in G is equal to the number of literals in the given CNF each vertex of the graph corresponds to a literal of E.Now, once we have formed the vertices of V we have to find a edges of G.How do we join two vertices by an edge in G?Two pairs i, j and k,lcan be joined by an edge or an edge exist between these two pairs i, j and k, L.If i 0is equal to k that is the first component, should be differentand X i j 0 is equal to 0 of X kl.

That is it should be possible to assign the values such that, both can have value 1 that is that this corresponds to a literal in the ith clause and this corresponds to a literal in thekth clause.They should not besuch that 1 is y and another 1 is not y, that is 1should not be the negation of the other 1 is variable and another should negation of the variable.They should not be of that form,if they are not of that form and it can exist between them.

(Refer Slide Time: 39:20)

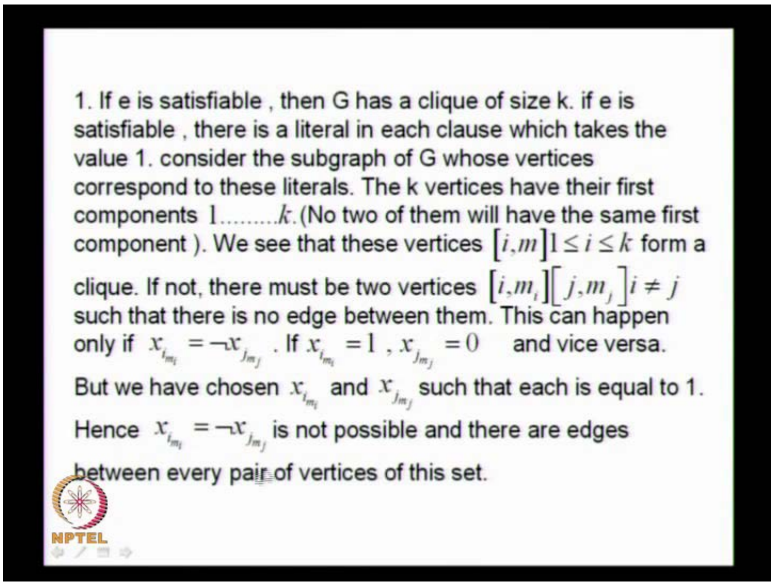


That is if 1 is, why we have this condition if 1 is y and the another is naught y, we cannot assign values independently.If we assign some value to y the other 1 automatically will take the other value, if we assign values to 1.1 the other will take the value 0 and so on, to enable independent assignment of values to X i j and X kl,we have the condition that X i j is not equal to naught of X kl.Now, how many vertices do we have?It is equal to the number of literalsin the expressionand number of vertices in is G is less than the length of e.

Because econsist of the literal cell has some more symbols right and the number of edges will be at most the squaredof it.So,G can be encoded as a string whose length is bounded by a polynomial in the length of e the number of vertices in G will beequal to the number of literals and hence less than the length of eand the number of edges at the most could be of the order n squared.

Where n is a number of verticesso, in if you want to encode G it can be encoded in such a way that the length is bounded by a polynomial in the length of e.And this can be returned down in time boundedby a polynomial in the length of e that is you have a polynomial time algorithm which will convert1 instance of CNF to 1 instance of the clique problem.Now, we show that G has a clique of size k if and only if e is satisfiable.

(Refer Slide Time: 41:21)

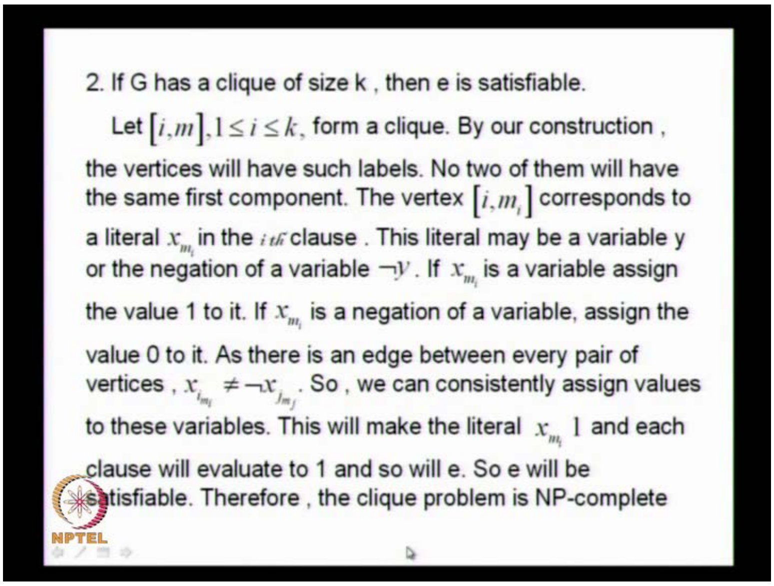


Now,if e is satisfiablewe have to show that G has a clique of,G has a clique of size k and vice versa.If e is satisfiable then every has, every clause has a literal which takes the value 1, that is a literal in each clause which takes the value 1.Now, you construct a graph from e and in this you consider the subgraph of G,whose vertices correspond to the literals, which take the value 1in each clause.We have at least 1 literal taken the value 1 take 1 literal from each clause, which has the value 1 considered that the subgraph of G corresponding to those vertices.

We will find that it is a clique it is a complete subgraph,the k vertices have their first components as 1 to k.No two of them will be the same and we considered the vertices i, m where i varies from 1 to k, this these vertices from a clique suppose, there does not existedge between two of them i, m,i j, m j, i and j are different,the edge will not existor this can happen only if x i, m i is equal to naught x j, m jthat is 1 is the negation of another that is 1 is a variable and another is thenegation of that variable.

So, if Xi m i is equal to 1 x j m j will be 0 or the other way around, but we have chosen the literals in such a way that, all of them have values 1from each clause,we have chosen a literal, which has the value 1.So, this cannot happen that is X i m i equal to naught X j m j is not possible.So, between every pair of vertices we have an edge.

(Refer Slide Time: 43:54)

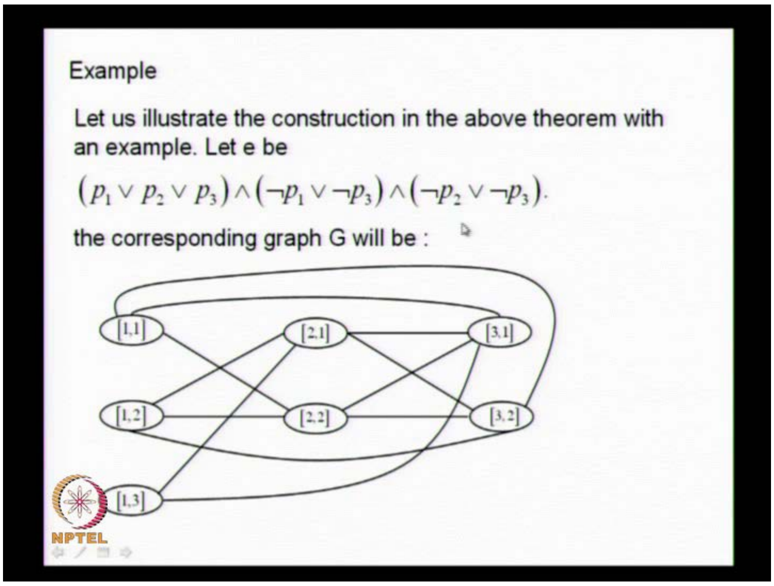


Now, the other way around if G has a clique of size k.Then we show that e issatisfiablelet i, m be a clique that is i should be all differentby our construction, the vertices will have such labelsand no of them will have the same first component, the vertex i, m i corresponds to the literal X m i in the ith clause.And this literal may be a variable or it may be the negation of a variable, it may be of the form y or naught y.If it is a variable assign the value 1 to it, if it is a negation of the variable assign the value 0 to it 0 to that variable as there is an edge between every pair of vertices X i m i is not equal to naught of X j m j.

So, consistently we can assign values to the variables because 1 is not the negation of another.So, consider the vertices i, m ithis will correspond to the literal X m i in the ithclause and if it is a variable, make it y if it is the negation of the variable make the variable 0.If we assign values to the variable this waythen each,clause will evaluate to 1 and so the expression e will evaluate to 1.So,e will be satisfiableso, what we have done is from 1 instance of CNF we have constructedan instance of a graphsuch that, the CNF expression is satisfiable if and only if,the graph has a clique of size k that is we have reduced CNF satisfiability to the clique problem.

And this transformation has been done in the polynomial time.So,polynomiallywe have reducedthe CNF problem to the clique problem in polynomial time and so the clique problem is NP-complete.So, this way known NP-complete problems can be reduced to unknown problems in polynomial time.And the new problems can be shown to be NP-complete, this is themethod followed to prove that some problem is in NP-complete.

(Refer Slide Time: 46:41)

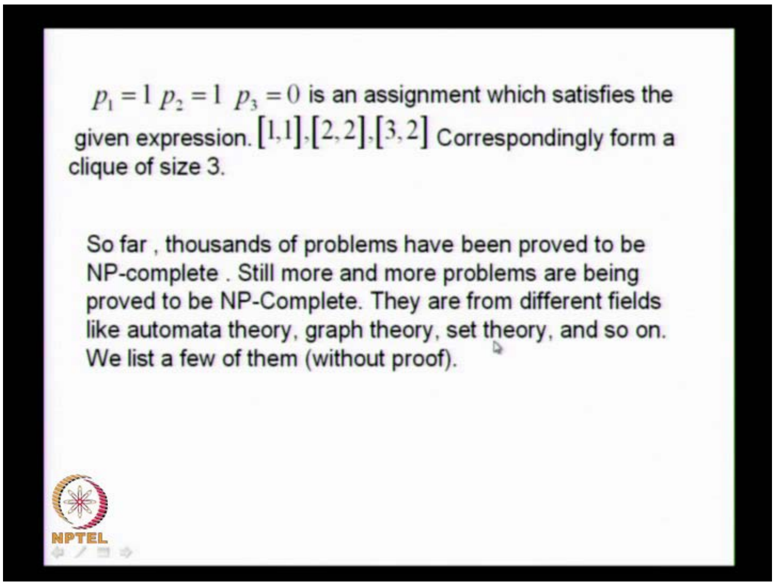


Let us illustrate the construction by an example suppose, the CNF expression is this p 1 or p 2 or p 3 and naught p 1or naught p 3 and naught p 2 or naught p 3corresponding to eachliteral,we have a vertex.So, for these 3, we have these 3 verticesthe first component tells you that it belongs to the first clause.Now, there are two literals in the second clause.So, we have two vertices there are twoliterals in the third clause so, two vertices with first component three.Now,from p 1 there will be an edge to every literal corresponding to this or every vertex corresponding to the literal unless that is naught of p 1.

If p1 is there if it is naught p 1 there will not be an edge between this and this.So,the you defined that there is no edge between this and thisso, there will be an edge between this and this, that is this and here p 1 does not exist at all.So, from p 1 there will be an edge to this, there will be an edge to this,that is from this there is an edge to this and there is an edge to this similarly, from p 2 there will be an edge to both this so, from this there is an edge here, there is an edge hereand here.

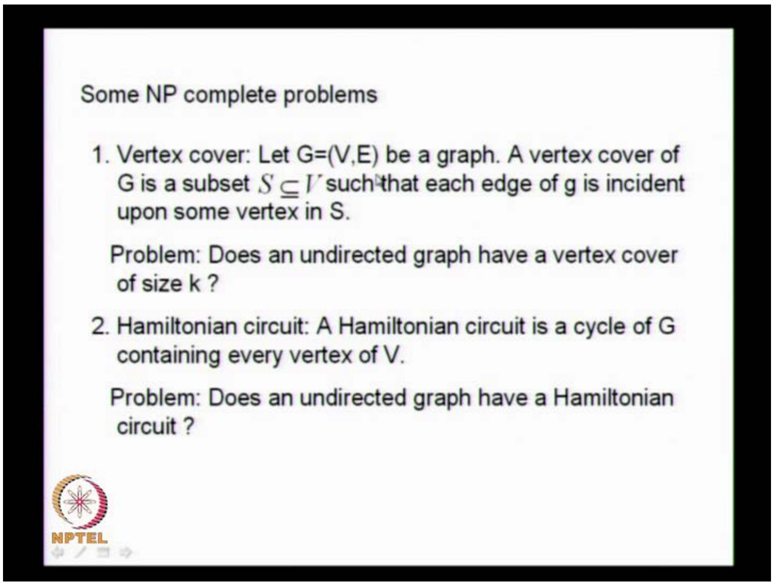
We have naught p 2.So,there will not be an edge between this and this there is no edge, but there will be an edge between this and this. This is given by thislike wise the address have been drawn.Now, you see that if p 1 is equal to 1 this is an assignment p 1 is equal 1,p 2 is equal to 1,p3 is equal to 0.1takes the value 1 this is evaluates 1,p 3 is 0,naught p 3 evaluates to 1.So,this is 1again p 3 is 0,naught p 3 evaluates to1.So, this clause is true.

(Refer Slide Time: 49:10)



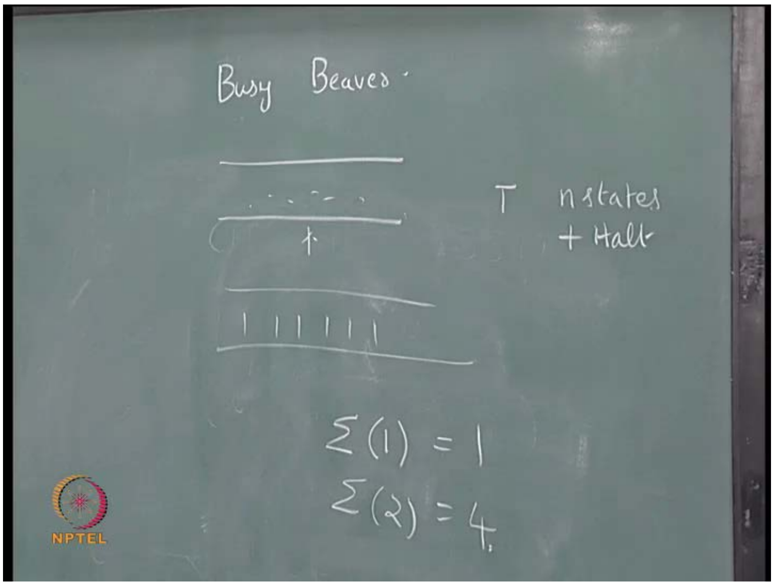
So, you find that thisand this form a clique of size three this.So, for thousands of problems have been proved to be NP-complete more and more problems are being proved to be NP-complete.They are from different fields some are from automata theory, some are from graph theory,some are from set theory and so on.

(Refer Slide Time: 49:26)



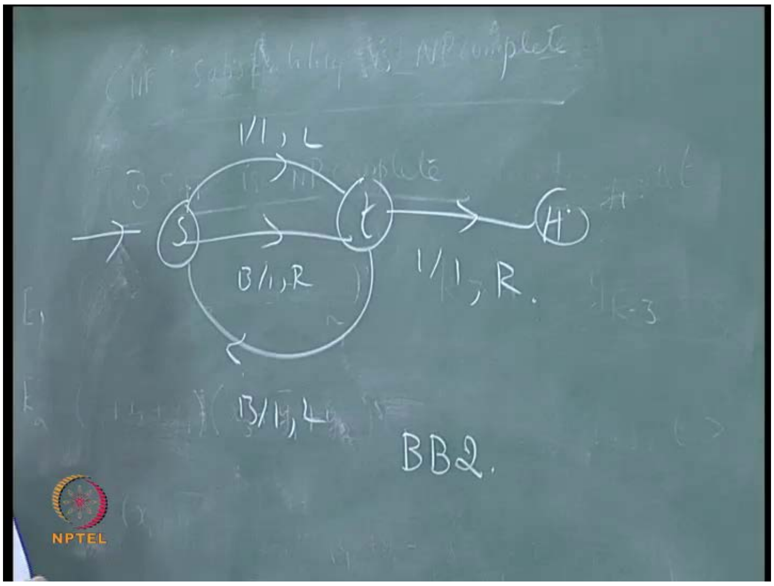
Some of them are vertex cover problem in a graph,Hamiltonian circuit problem in a graph,set cover problem in set theory, regular expression inequivalence in automata theory.The three dimensional matching in set theory integer programming problem these are some of the problems, which have been proved to be NP-complete.So,I will just deviate andtalk about the Busy Beaver problem which I give you.

(Refer Slide Time: 50:07)



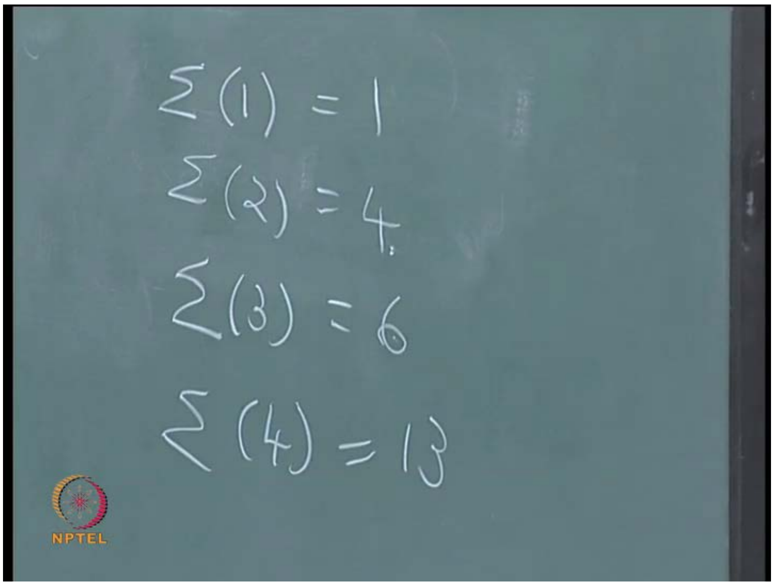
What is a Busy Beaver?There are slightly different versions given in different book.So, when you read a bookit is slightly different you,I mean you may feel that, but the original problem is givenlike thisconsider a Turing machine withn states and a halting state,n non-halting state and final halting state.If started on a blank tapewhat is the maximum number of 1’s it will print and then halt,that is theknown as this sigma is given by a function sigma 1,sigma0 will be 0.Sigma 1 is 1sigma 2is4 the machine for that.

(Refer Slide Time: 51:53)



We can have.(( ))Two statess and t this the initial statethen a halting statethis machineis the solution forBB2,the assignment I have asked you to construct BB3.

(Refer Slide Time: 52:36)

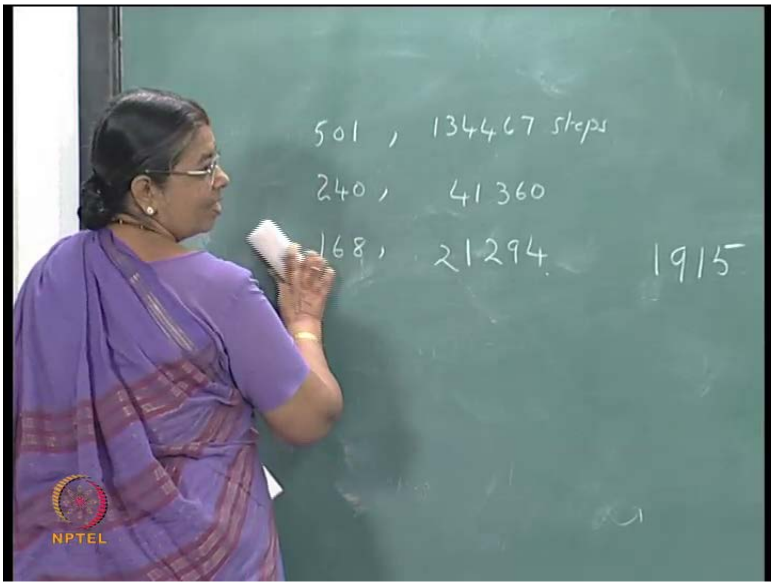


The value for that is sigma3 is 6,sigma 4 is13,sigma 5 start a computable function it is very difficult to find out.What is the value of sigma 5?Sigma 5so, people we have to try out all possibilitiesand seewhat is this in fact in some books, the variation is slightly its slightly different the problem is specified asthe tape alphabet is taken at0,1 and blankyou start with 11 on the tape n 1’s and the machine has nhalting states plus 1 more statesthen, ultimately when it halt, how many what is the maximum number of 1’s that is print slightly different version of this, but it computes the same function.right

Now, at one time there was a competition held forI mean, find out.What is the value of sigma 5?This is a somethingGA conference which was held in (( )) in 1982 or 83andthey were given prizes(( ))have to run a program and checkthis,these are something like Oscarprizes know,which the Beaver.What is a Beaver?Beaver is something like,antwhich brings the twigs and builds something.So, the prizes were given as in terms of a statuteits look likes a beaver.

So, this first prize, second prize,third prizewere given.So, the solution at that timewasnow, this sigma 3 is 6, but that machine makes 13moves.Number of moves which it makes for printing is also important.

(Refer Slide Time: 55:04)

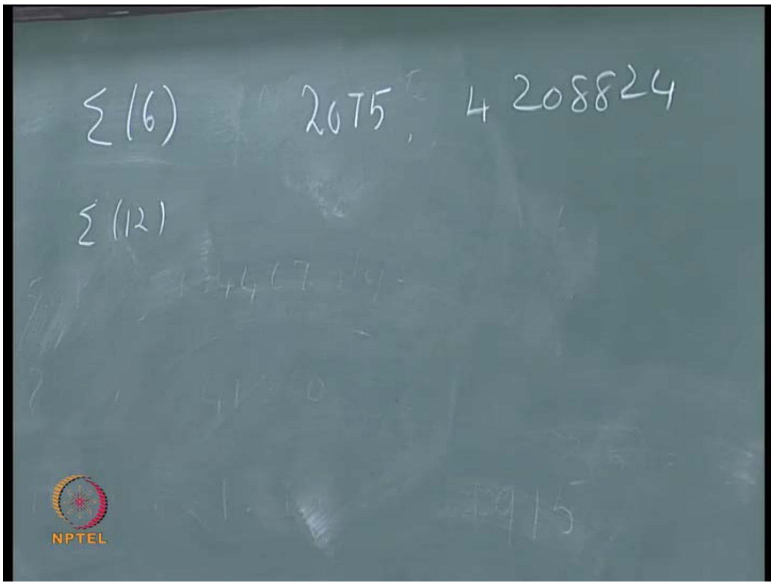


So, in that the first prize was won by a person(( ))from Hamburg,he had5011sprinted in 134467 stepsand the second prize was2401sprinted in41360 steps and the third prize went to onessecond prize(( )) from BadenSwitzerlandand another one is third prize is ((

))168ones with21294 steps,but that is not the final one later on 84 or 85 with 5 statesso, many ones were printed.Ido not have the correct reference, butDuedney1984or 85there were 1950ones were printed with 5 states.

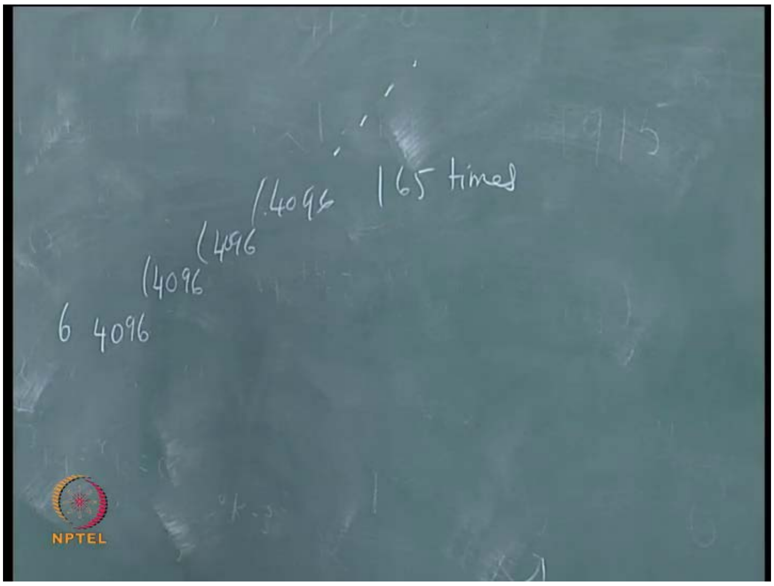
It is not compute if the function isgrows very fast, it will notcomputable,like you know Ackermann function is computable it is a not primitive recursive because it grows very fastcompared to any primitive recursive functions here.

(Refer Slide Time: 56:57)



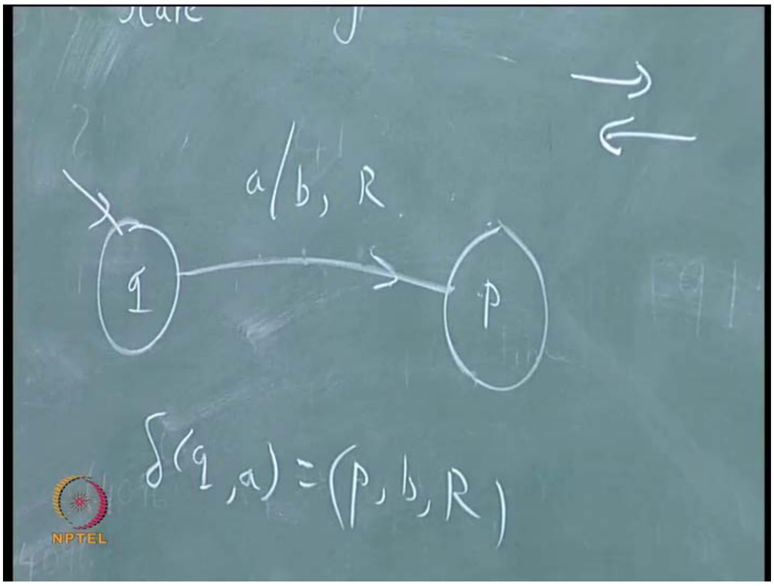
So, sigma 5 has so many values.What about sigma 6?At in 83 itself there was a machine which could print 2075 onesin4208824 steps, later on it might have been improved atsee at that time in 83 or so the interest was so, much on this finding out similarly, at one timetrade off between the states and the symbols.State symbols of Turing machine was of verybig importance people,many people were working at that time nowadays nobody works on thatI think so, then sigma 12apparently it is more than this at least there is one machine.

(Refer Slide Time: 57:58)



Whichhas so, many one’s printed 6 times4096to the power of 4096 to the power of 4096 to the power of4096like that,this is 165 timesin 83they had a machine which could do this.So, it may be larger than this so, for twelve states itself number is very largethat means, you can see that it is not a computable function it is very difficult to compute.So, this is a just a (( ))and regarding the state,Ido a diagram of about state diagram of Turing machines.

(Refer Slide Time: 58:57)



State diagramof Turing machines can be like this,from one stateyou go to, another statethe move can be written like this.That is delta ofq, a is equal top, b,R the mapping can be written in this manner sometimes instead ofR, R, L this symbol else also used to denote R,R, L,but this is in the recent books, recent book meansbooks written aftersay 87,88 the earlier books. Bookswhich were written in the 70'sself loops will be usually omitted.

(Refer Slide Time: 59:49)

And the state will bethe, state name will be written hereand R, R, L will be written within the cell.Within the circle the earlier books, which are written in60s and 70s the state diagram is denoted like this.Andthis statethat means from q 0 after reading a you rewrite it with b and go to q 1 and move left.

When you go to q 1 you move left,when you go to q 0 you move right,this is the way Ithink is written and usually,if you just do not rewrite, but move right in the same state or move left self loops will be there,though self loops are usually not given in the diagram that is theconvention followed in earlier books.Books which are written say the 70's and 60's.So, if you take a booksee.What notation he is following?So,with this I stop.