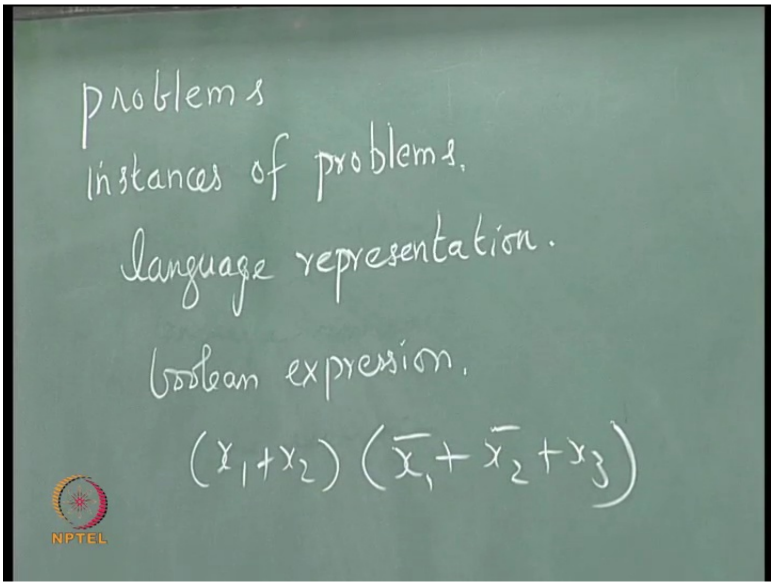
Theory of Computation

Prof.Kamala Krithivasan
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture No. # 33

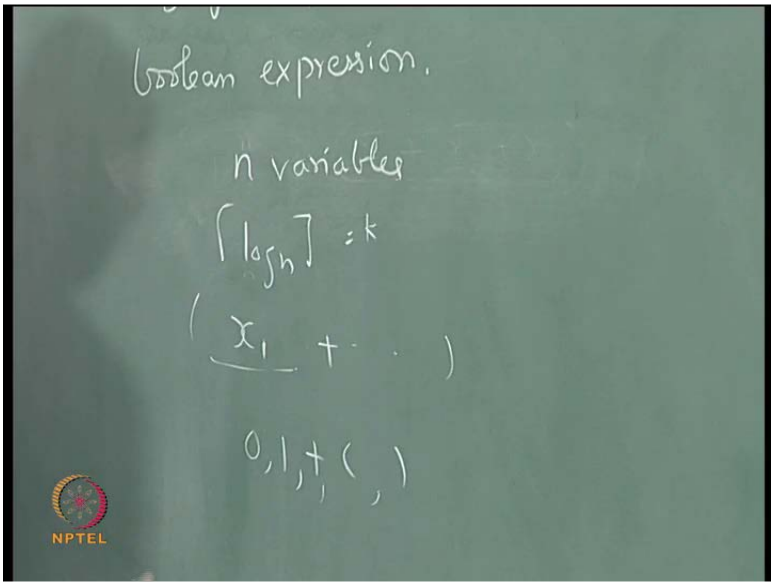
Rice's Theorem, Linear Bounded Automata, Properties of TM

(Refer Slide Time: 00:18)



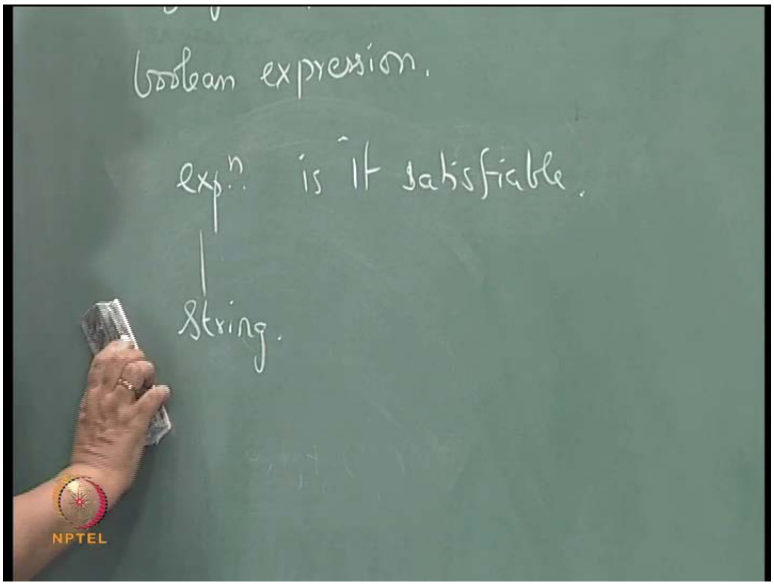
We were considering about decidability,and undecidability.So, we were considering problems,and instances of problem (No audio from 00:24 to 00:32)and what is a language representation?(No audio from 00:35 to 00:47) For example, consider a Boolean expression. (No audio from 00:51 to 01:00) Is it satisfiableor not? That is a decidable problem. Is it satisfiableor not, given an expression something like thatx 1 plus x 2,x 1 barplus x 2 bar plus something like that. This is a Boolean expression, is itsatisfiablemeans is there an assignment to the variables which will make the expressionequal to 1.This is a decidable problem,we can definitely find out whether it isthe satisfiableor not, butwhatthere are several instances of the problem. What is an instance? An instance is an expression, one expression is an instance.

(Refer Slide Time: 01:50)



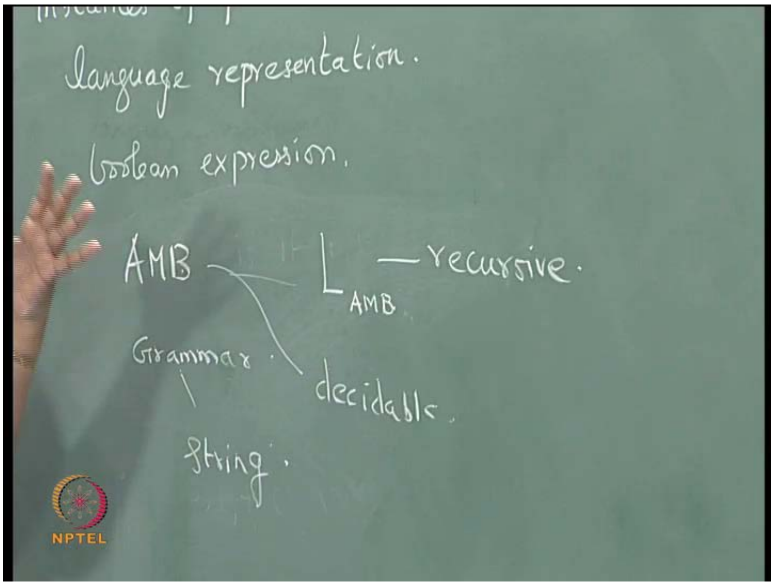
So, you can have a Turing there; as a language you can represent this in some manner. For example, each variable you can represent as 10 powern means x n or something like that some representation oryou can have sayif there are n variables. We can uselog nbits to represent one variable,and so on.So,one variable you can use usingsay some k bitsand then plus you can use brackets you can use. Like that you can use the symbols,01 for representing this plussymbol then parenthesisand so on. With these symbols you can write a stringand you can have a Turing machinewhich willaccept those strings which evaluate to 1 and which will not accept those strings which evaluate to 0. Like that you can have a Turingmachine.

(Refer Slide Time: 02:53)



So, this is a decidable problem, butas a language accepted. See theproblem isgiven an expressionis itsatisfiable? As a language representation how would yourepresent it? Assess thisexpression is represented as astringand you can have a Turing machinewhich will accept those strings which evaluate to 1 and which will reject those strings which will notevaluate to 1.Now,the ambiguity problem again we consider,ambiguity problemthat isgrammar is the inputand is it ambiguous or not?

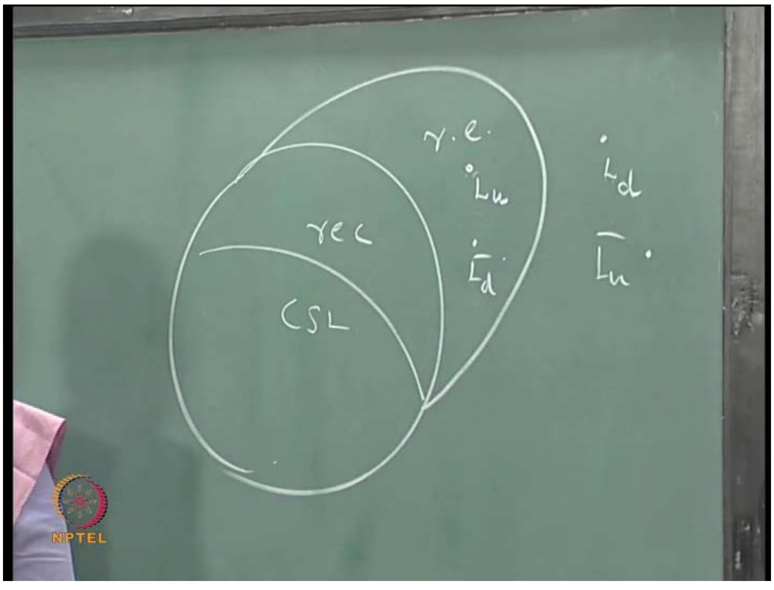
(Refer Slide Time: 03:35)



Now, the instance of the problem is one grammarand this one grammar you can represent as a string. Using some encoding for Turing machine like that grammar can be represented as a string. If it is decidable there will be Turing machinewhich will accept all those stringswhere the grammar isambiguous and they reject all those strings which are not ambiguous.So, there will be athe corresponding languageI cancall as L AMB. This consists of stringswhich represent grammars which are ambiguous. Now, when do you say that ambiguous is decidable?

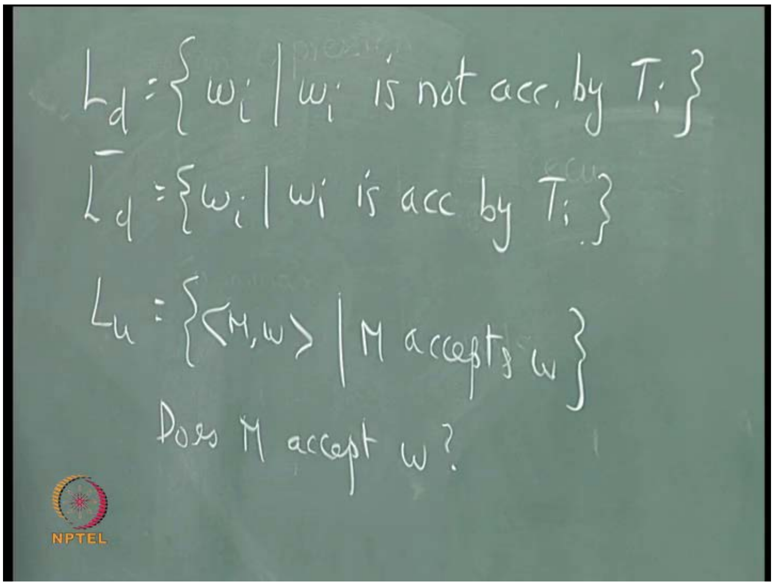
When do you say it is decidable? When this language is recursive. It has to halt and say;when do you say that a language is recursive? It corresponds to a Turing machine which halts on all inputs. So, it has to ultimately halt and say yes or no. So, here any string has to the machine has to halt andaccept or reject. Then it isdecidable. If it accepts and stops, butwhen it rejects it gets into loop means that is not decidable. The language has to be recursive. You have a problem and a corresponding language representation. The problem is decidable if the corresponding language is recursivenot otherwise.

(Refer Slide Time: 05:28)



Now, there are two casesif when it is undecidable? So, this is recursivecontext sensitivewill be hereand this is recursivelyenumerable,this is the hierarchy.So, if a problem is decidable the corresponding language has to fall within this.Now, if it does not fall within this it is undecidable problem. If it is undecidable, it may be here or it may be here that. For example, this L u,we have seen some examples here L u is here, L d is hereisnot it.

(Refer Slide Time: 06:24)



We have seen what is L d diagonal language. What is L d?(No audio from 06:14 to 06:23) L dconsists of strings w i;w iis notacceptedby the i thTuring machine. We know that the Turing machine can be encoded using 0 and 1and itthey can enumerated. L d bar is thecomplement of that isw i;w iis acceptedby T iand L u isM W;M accepts W,universal language.Now, L u and L d bar are here. L d is here, L u bar has to be here. The reason isif a language is recursive is the complement will be recursive. If a language and it is complement are both recursively enumerable then it can recursive, it has to be recursive.

So, the thing is L u is here L d L u bar will be here, L d is here,Ld bar is here. The possibilities is for a language L and L barif both of them can be within this or both of them can be outsideor one here, one here. These are the three possibilities and this we have already seen. So, problem is decidable only if the corresponding language falls within thiseven if it falls here or here. It is undecidable.So, the corresponding problem L for L u is,doesM accept W? This is the problem the language representation is this. Thisproblem does acceptsM? M accept W is undecidable. Halting problem is,does Mhalt on W? Slight difference is there,I meanthere is a difference.

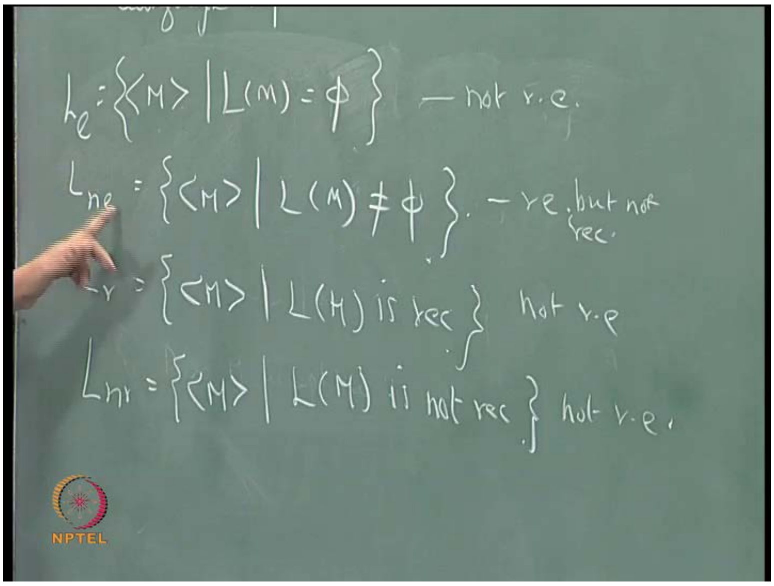
This is NM accepts W. Now, we have seen two ways of proving the halting problem. All most similar idea, butin one we showed that if the halting problem were decidable thenyou will have aTuring machine for L d. That is not correct.(No audio from 09:00 to 09:06)Now, we have seen that L u is recursively enumerable, butit is notrecursive because we have seen that if L u isrecursive then we can have a Turing machine for L d bar.Now, there are several problems which aresometimesthe word partially decidable is used if the language falls here. Language isrecursively enumerable, butnot recursive. Then the word partially decidable is used. It is not a very common thingin books on logic that word is rather used.

For example, the validityproblem of proportion logicwhat is the validity problem for proportion logic? The validity problem of proportion logic is similar to the saidproblem?In the validity problem we try to find out whether a formula in proportional logic is a tautology? That is whether it evaluates to 1 always. In said problem we try to find out whether there existsan assignment to the variableswhich will make the Boolean expression evaluate to 1.It is somewhat we can see the correspondence between that andsatisfiability of the Boolean expression. We have taught,you have learnt about tautologiescontingencies and contradictions.

When is it a tautology? You have to draw the truth table and for last column you have to look into that. If it is all 1 it is a tautology, some 0’s someone it iscontingencies,all 0’sit is a contradiction. You can always findthat draw the table and find out is not it. So, it is a decidable problem. What is a validity problem for first order logic? Given an expressionis it valid in the first order logic; is it valid or not? Like for all x for all y,p of x y or something like thatis equivalent to saying for allx for all y is equivalent to saying for all y for all xbecause if both are universal quantifiersyou can interchange them. Both one is existed you cannot interchange. Some expressions like that you can say that they are valid expression.

Is there a way of finding out whether a first order logic expression isvalid or not? Lot ofI mean work has beengone in that direction.So, for first order logicthe language will be here, for second order logic it will be here. What is second order logic?Even the proportional things could be quantified. So, in first order logic individual variables can bequantified using they can be bound using quantifiers. Insecond order as even predicate variables can be bond using quantifier for all p, p where isp representsa predicate and so on.

(Refer Slide Time: 12:40)

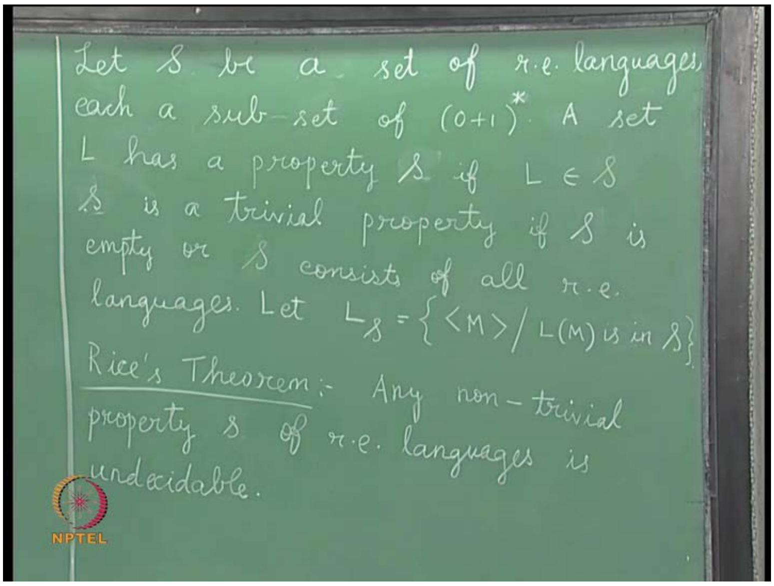


So, there areresultslike that, butwe will not go into them. Some certain things like see saygiven M is an encoding of theTuring machine L eisM; L of M is equal tophi. Does the language generate the empty set? L non-emptyis L of Mnot equal to phi. L rall Turing machines encoding of Turing machines L of Mis recursive. L n rM encoding of Turing machines L of Mis notrecursive. Some languages the corresponding problem is givenTuring machine is the language generated empty. Given a Turing machine is the language generated non-empty. These are the problemsgiven a Turing machine M is L M recursive. Given a Turing machine M is L M non-recursive.Input is the encoding of the Turing machine these areall some problems.

Instances of the problem will be particular Turing machineand you are given the encoding. You have to find out whetherit will,it accepts the empty language or is the language accepted non empty. Is thelanguage accepted recursive. Is the language accepted non recursive and so on.The corresponding language representation is this. For the language representation the encoding of the Turing machine is the string which has to be accepted or not accepted. Now,this L n e falls here. It isr e, but notrecursive, but these things arenot even r e. They are all undecidable; they are all undecidable problemsbecause when do you say it is decidable?

When the language is recursive, buthere the languages are not recursive, but this alone is recursively aenumerable, but not recursive it falls here, the other three fall here.So, the proofs forthem each one you can give a proof. The argument will beall most similar to what you consider for L d, L d bar etcetera. There will be slight variations, but the arguments will be somewhat similar. So, I am not going into thedetails of that. In factthere is a general result like this rice's theoremfor recursive sets.

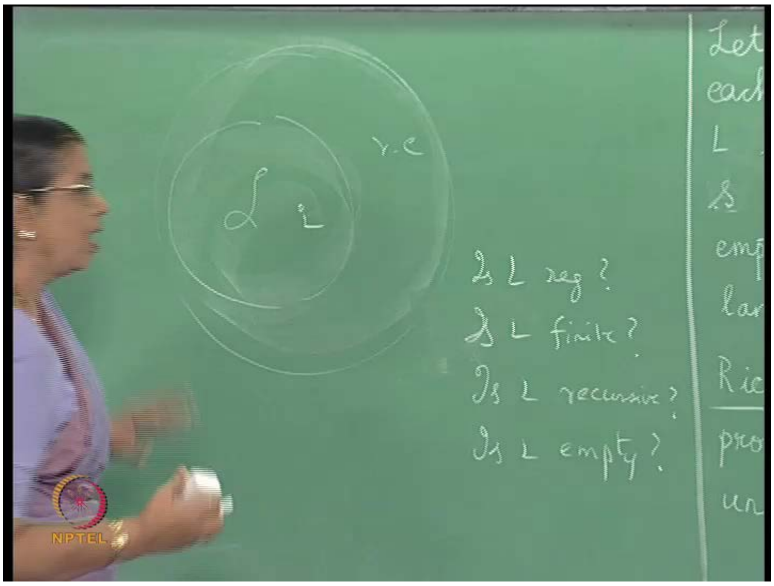
(Refer Slide Time: 15:43)



There is another theorem forrecursively enumerable index sets that we may be a may not be able to consider, but this one we will consider. What is rice's theorem?Let s be a set of recursively enumerable languageseach a subset of 01's with 0 plus 1 star. That iswithout loss of generality we assume that the alphabet consists of only 0 1and blank. Input alphabet is 0 onlyapart from that we have the blank. Actually for any Turing machine you can just have only 1 and blank;0 representing the blank and 1 symbol. That is also possible, but a number of movesto simulate any Turing machine with suchalphabet. That is 01 where 0 represents the blankwill be enormous,number of moves required is enormous and that is whywe are not considering it.

For our proves it is enoughif we considered that the tapedalphabet to consist of 0 and 1and a blank. Blank is separate 0 1 because any symbolyou can encode in binary. That is an idea,that we already seen.

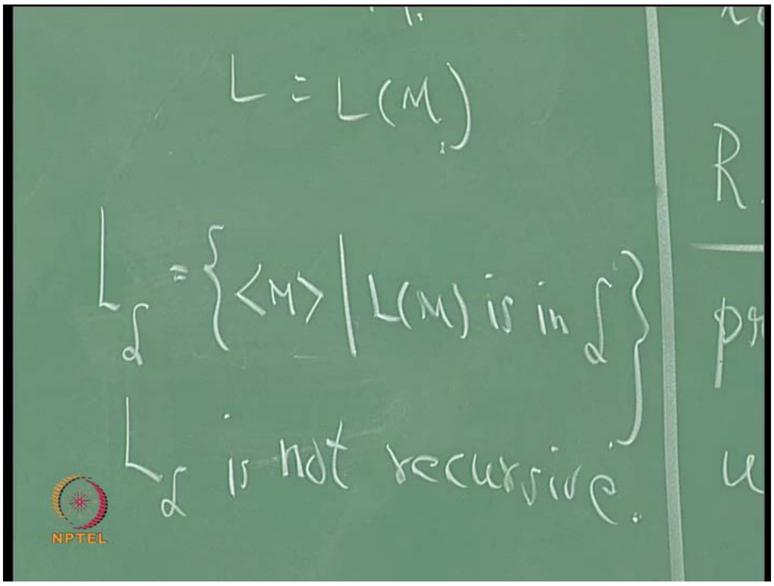
(Refer Slide Time: 17:20)



So, without loss of generalitywe assume that the input alphabet is 0 1and tape alphabet has 0 1 and blank. And s be a set of recursively enumerable languages. So, the whole family of recursively enumerable languages is this means you have considering asubset sof recursively enumerable languages. Now, a language L has the property s. If L is here, L has property s. You say L has property s. A set L has property s if L belongs to s. s is a trivial property if s is emptyor s is equal to the whole family recursively enumerable languages.s is a subset of r e.

Now, if s is equal to the wholesetor if s is empty. It is a trivial property. Otherwise it is a non-trivial property,Is L s regular?Is s finite? I am sorry. Lyou have to sayis L regular?Is L finite?Is L recursive?Is L empty?All these are non-trivial properties. They are all non-trivial properties. Rice’s theorem says states that any non-trivial propertysof recursively enumerable languages is undecidable. Whatever property you take if it non trivialit is undecidable.(No audio from 19:10 to 19:19)

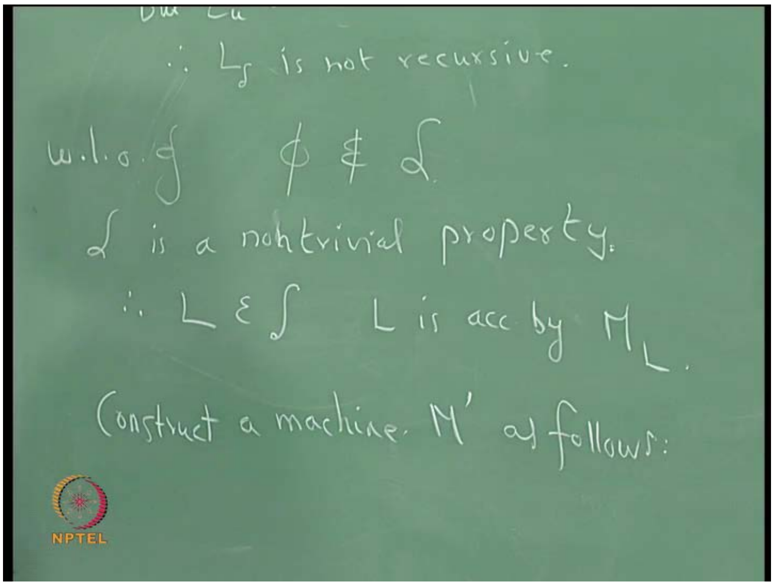
(Refer Slide Time: 19:42)



So, L of s the languages representation for s is the encoding ofTuring machine such that L of M s in s.(No audio from 19:30 to 19:38) See L r e languages are this, L is a, s is a subset of thatthis is s.Now, L is in s means L has property s. L will be accepted by a turing machine M, L is a recursively enumerable set. So, it has to be accepted by a Turing machine M. So, L is equal to L ofM.So, if L of M is in s the encoding of M take the encoding of M. That is M;L sis a languagewhich representsthe encoding of Turing machine L of Mis in.

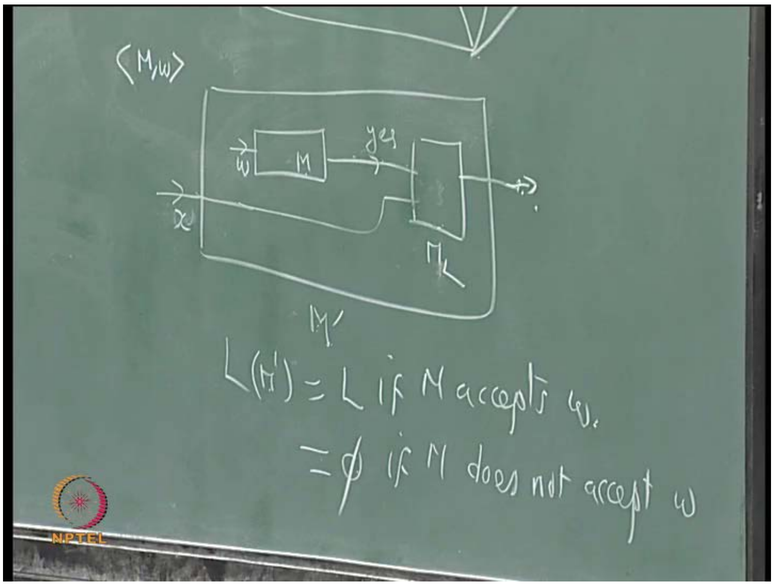
Now rice's theorem says that L of s is not recursive isnot it. Rice's theorem says that L of s is not recursive. There is another theorem which says under what conditions it will be recursively enumerable. It is not recursive meanssay Itold you could be recursively enumerable, butnot recursive or it may be here also. L of s if s is a non-trivial property, L of s will be here or here. In which case it will be here and in which case it will be here? That is again in another theorem forrice's theorem for recursively enumerable index setswe shall consider later. What we want to prove that L s is not recursive?(No audio from 21:41 to 21:52) How do you prove this?(No audio from 21:56 to 22:11)

(Refer Slide Time: 22:13)



So, you prove like thisif L s is recursive, L u will be recursive, but you know that L uis notrecursive. Therefore, L s is not recursive. This will bethe argument.(No audio from 22:50 to 22:57) Without loss of generalityempty set does not belong to s. Assume empty set does not belong to s. Otherwise you have to take s barcompliment of that does not matter. So, sis anon-trivial property.(No audio from 23:24 to 23:31) Therefore, there some L belonging to s. There is some language belonging to sand L is acceptedby a Turing machineM L. The corresponding Turingmachine is M L. L belongs to s, there is a language L belonging to s and L is accepted by a machine Turing machine call it asM L.

(Refer Slide Time: 24:37)



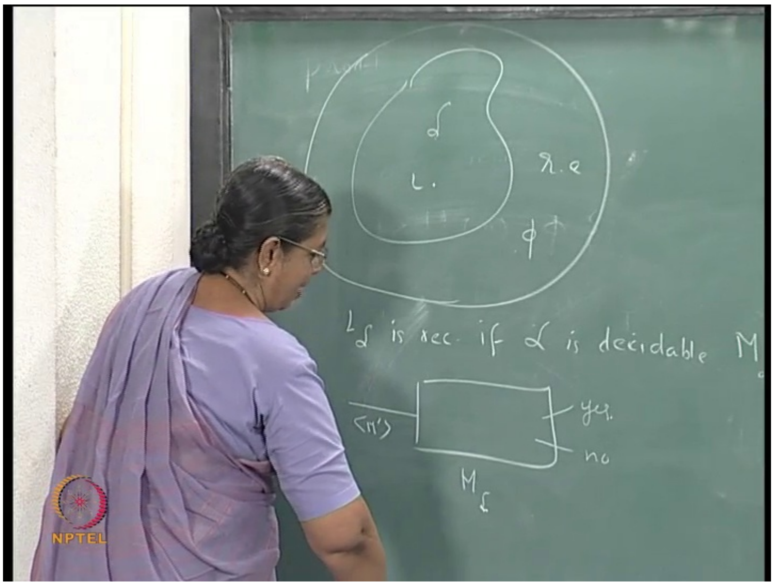
Now, you construct a machine,machine means Turing machineM dashas follows. How do you construct M dash?(No audio from 24:24 to 24:35) The structure of M dash will be like this.

(No audio from 24:38 to 25:15)

What M dash will do is it will stimulate the behaviour of M on w. It will ignore the input x.M for M dash the input is x it will ignore that, but it will simulate the behaviour of M on w. And if the answer is yes, it will start M L and thenfor every M w you can construct a machine M dash as follows, for every pair M and wyou construct the machine M dash like this.What M dash will do is it will ignore it is input , but it will simulate the behaviour of M on w. And if it is yes, then it will start M L. If it is no,then M L will not be started. If M L will be started and it will work on xand if x is accepted by M L,it will say yes. If it not accepted, it mayget into loop or reject. If it is yes, it will say yes.

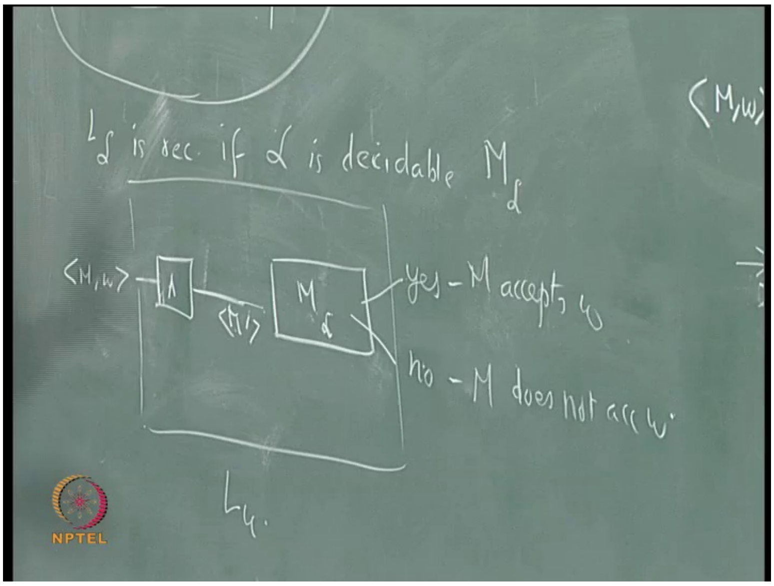
So, what is thelanguage accepted by M dash? L of M dash,if Maccepts w then this will be startedand if this is started, it will accept L.M L will accept only the language L is not it. M L is the Turing machine for L so, L of M dash is L if Macceptsw. For every pair M w you can construct a machine M dash like this. It will M dash ignore it is input initially. It simulates M on w and if the answer is yes,M L is started and it works on x and if the answer is yes, it will say yes.So, if M accepts w only this will be startedand itwill accept just the language L. Whichever string is in L it will accept whichever string is not in L it will not accept.So, L of Mdash will be L if M accepts w. L of M dash is equal to empty. If M does not accept w,this will never get started. So, the language accepted will be empty. L of M dash is empty,if mdoes notacceptw.

(Refer Slide Time: 29:02)



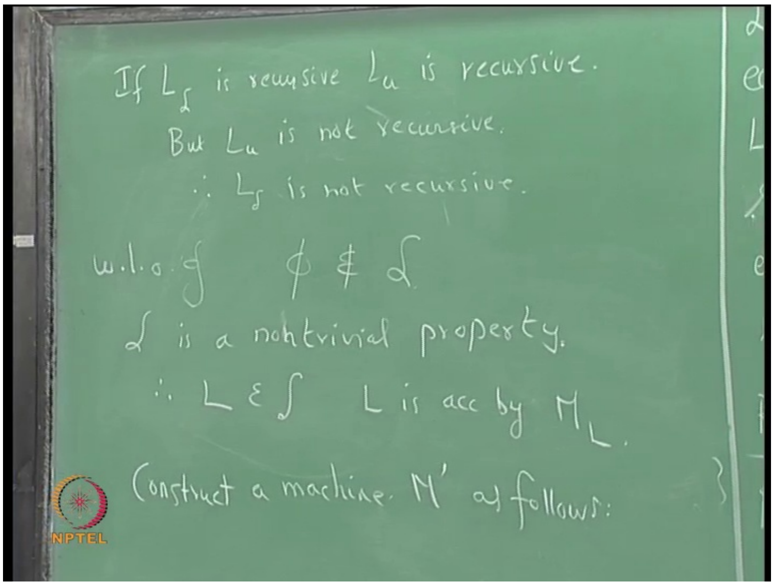
Now, (No audio from 28:47 to 29:00)S is a non-trivial property. If this is thefamily of r e languages,s is here and we have seen that phi is not in.It is not inphi is here, L is here. Now, suppose s is a decidable property. Then what can you say about L s?It has to be recursive. L sis recursiveifs isdecidable. So, there will be a Turing machineM s for that.Now, this is the Turing machineM s. For thisgive the M coding of M dash. If it says yes,what does that mean? If it says yes means L M dash is L. When is L M dash L? If M accepts w. So,M accepts w. If it says no,then Mdoes not accept w. So, you are able to say whether m accepts w are m does not accept w that is L u is recursive, but L u is not recursive. I will write thisdiagram in a slightly different manner.

(Refer Slide Time: 30:56)



So, givenM wyou have an algorithmwhich will construct M dash. This construct m dashand the encoding of M dash is given asan input toM s(No audio from 31:22 to 31:28) and if this says yes,that meansM acceptsw. If this sayssee we are assuming thatthis isrecursive and those there is Turing machine which halts on all inputs. L of s is recursive if L s is a decidable property. So,there is a Turing machine M s which halts on all inputs. So, it willsay yes or no,M does notaccept w(No audio from 32:08 to 32:15) that means you are having an algorithm for L u or you are having asaying that the problem with a thus M accept w is decidable.This is an algorithm for L u or a Turing machinewhich halts on L on all inputs, which is a contradiction. We know L u is not recursive. If you assume that this property non-trivial property is decidable then you come to the conclusion that L u recursive.

(Refer Slide Time: 33:03)

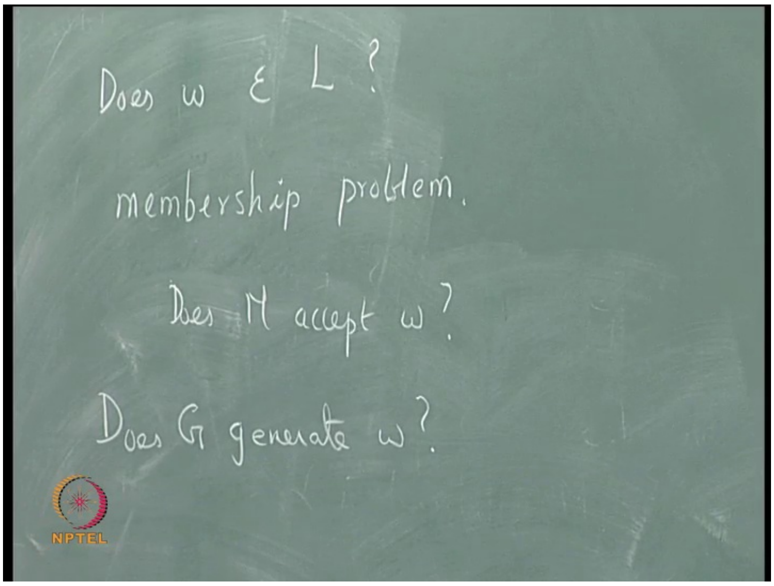


So, the argument is like this if L s is recursive then L u is recursive. That is what we have come to the conclusion, but we know that L u is not recursive,that is a contradiction. So, L s is not recursive or s is not decidable. This is for any non-trivial property. So, any non-trivial property of recursively enumerable language it is undecidable.(No audio from 33:25 to 33:30) So, this sort of an argument initially it may look a little bitnot very convincing or somewhat confusing. If you read the argument again and again it will become rare. I am doing only this theoremrice’s theorem only. If you look into the proof from the book L empty, L non-empty, L recursive all the proofs will be similar. There will be slight difference in each one them. That iswhy I said we to be very clearas to what want to prove and so on.

So,under what condition, we know that L s is not recursive, but underwhat condition it is recursively enumerable? That is another theoremunder what condition it is not even the recursively enumerable? That is you can find it.(No audio from 34:26 to 34:38) So, as I mentioned earlieris L empty?Is L recursive? Is L non-recursive? They are all thecorresponding languages fall here is L non-empty fallshere. Similarly, is L finite? Is L infinite? Is L regular?Does L have at least ten elements? There is a particular member belong to L. All those things are non-trivial properties.

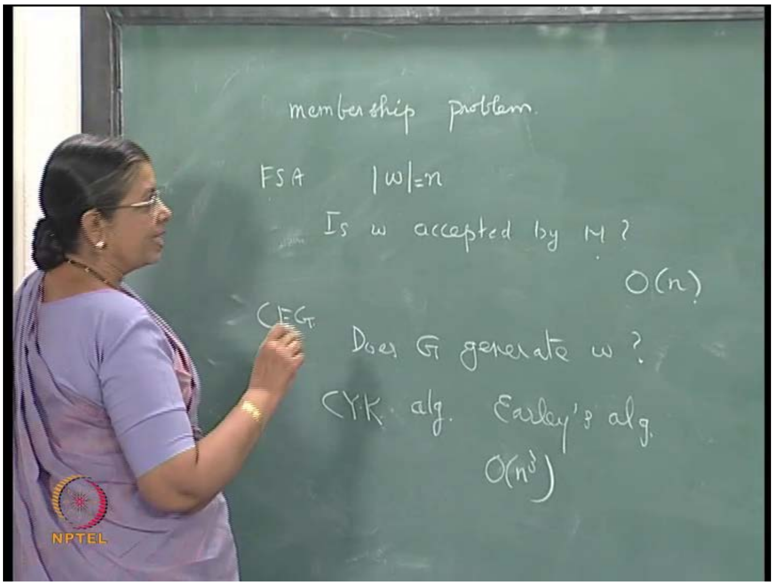
Trivial property is when s becomes the whole set or the empty set.Is L recursively enumerable? That is a trivial property because it is s for all recursively enumerablelanguages. So, whenever you find that there is a non-trivial property of recursively enumerable language. You can need not even think about this detailed proof,immediately use rice’s theorem and say it is an undecidable.

(Refer Slide Time: 35:50)



So,doesw belong to L or given a type 0 grammarwill it generate astring w. That is an undecidable problem. Let us call the membership problemor for a Turing machine corresponding Turing. DoesMacceptw? As a grammartype 0 grammar you should take. Does ggeneratew? If g is type this is undecidable. As a Turing machine we know that this is undecidable. Whereas, the membership problem for contact sensitive languagesis decidable, context free languages it is decidable for finite state automata or type 3 languages it is decidable. And the question of how much time it will take to find out whether w can be generated by a grammar? If it isright linear,if it iscontact free and so on.

(Refer Slide Time: 37:32)



So, if you look into the membership problem.(No audio from 37:31 to 37:41) For F S A,DoesF S Aaccepts a string w? How much time it will take?Functionas a function of the length of the input. w is w accepted by a finite state automaton. The length of wis equal to n. Is wacceptedbyMwhere M is aF S A?How much it will take? You have to just look into that. So, norder n time. Membership problem for F S A you can solve in linear time. For C F Gor C F L it isdoesg generatew?This is the membership problem for context free languages. There are algorithms likeC Y Kalgorithm,Cocke–Younger–Kasami algorithmand Earley's algorithm(No audio from 38:58 to 39:05) which solve this problem in order n cube time.

Somes manipulation you can reduce it to exactly less than 3 like matrix multiplication. You can reduce using stars and matrix multiplication, like that some L two point something you can reduce, but that is a verydifficultproof. Generally, you doing into order n cubed. In fact sometimes if the grammar is unambiguous,it will work in order in n square time. Earley's algorithm will work in ordering n square times if the grammar is unambiguous, but finding out whether a grammar is a ambiguous or unambiguous is undecidable problem.

(Refer Slide Time: 40:30)

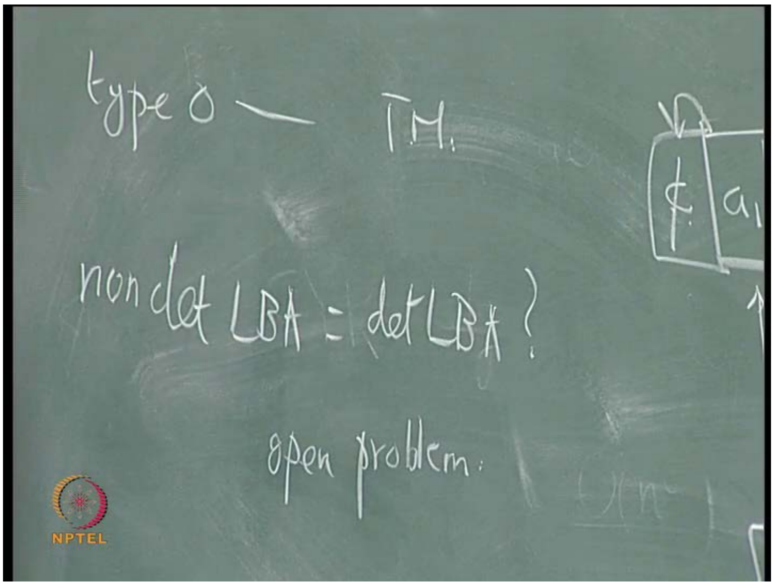
Now, for context sensitive language, membership problemis said to be p space complete. We will see that later. Another thing isfor type 3 grammars the automaton is F S Aand in F S A deterministic and non-deterministic version are equivalent. Type 2grammarsC F Gthe automaton is pushdown automaton. Type 0 it is Turing machine.Here the non-deterministic and the deterministic version are not equivalent. Here the non-deterministic versionis equivalent to deterministic version. As far as accepting power isconcerned, but the number of steps will be exponentially increased when you try to simulate a non-deterministic machineTuring machine with a deterministic Turing machine.

What is theautomaton for type 1grammars? It is calledL B A or linear bounded automata. What is a linear bounded automat? It has a tapewhich is linearly bounded with n marks. You have a tape;the input will begiven on the tape. The machine will start here in the initial state. It in the behaviour is just like a Turing machine. Any time it can read a symbolin a state mapping will be just like this delta ofq ais equal topARor L. So, it will change it is state to p.Print a Amove right or move left;behave, mapping is exactly similar to Turing machine, but it cannotit has to move within this.

If it tries to reach this cell or this cell the input will bethe machine stops. If it reaches the n marks the machine halts. So, it has to move within this space only. Whereas a Turing machine infinite inboth directions ora one direction whatever it is one way infinite. Here the tape is finiteand behaviour mapping is similar tothat of a Turing machine, but if it reaches this cell or this cell machine halts. Within the end markerit has to move. It is linearly bounded; the tape is linearly bounded that is why it is called linear bounded automata.Here again, it has been shown that type 1 grammars are equivalent to non-deterministic linear bounded automata. We can give a proof saying thatgiven a non-deterministic linear bounded automata you can have a context sensitive grammar.

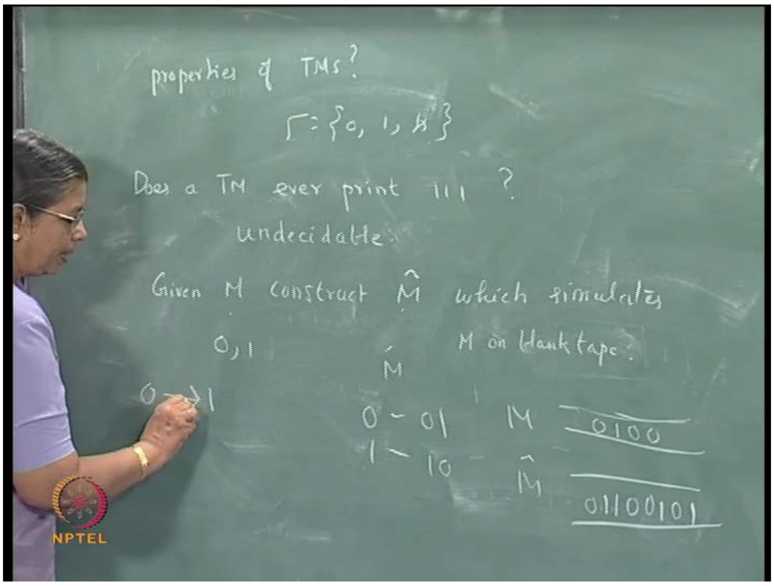
Given a context sensitive grammar you can have an non-deterministic linear bounded automata, but if you take any context sensitive language. You are able to constructadeterministic linear bounded automata whereas in the case of pushdown automata for w w r or certain languages,you are not able to construct the deterministic pushdown automata. Here whenever you take a context sensitive language you are able to construct the deterministic linear boundedautomata. Proof showing the equivalence uses non deterministic linear bounded automata, but so for any context sensitive language if you take. You are able to construct the linear deterministic linear deterministic bounded automata for that.

(Refer Slide Time: 44:20)



So, whether non deterministicL B Ais equal to deterministicL B Ais an open problem. Still now; till now it is an openproblem.(No audio from 44:41 to 44:47)Now, rice’s theoremsays there is any non-trivial property of the recursively enumerablelanguagesis undecidable. So, any property if you take you can very easily sayit is undecidable or not, but what about properties of Turing machines?

(Refer Slide Time: 45:15)



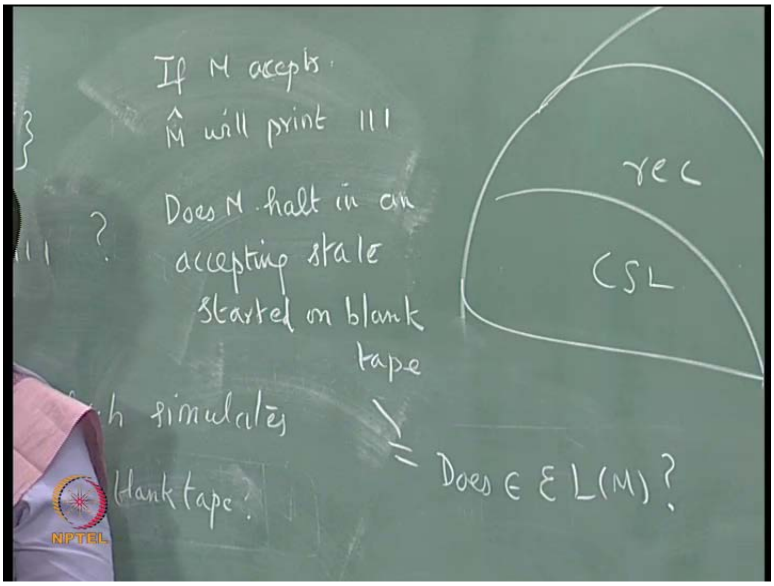
Propertiesof Turing machines,properties of the languages accepted by Turing machine is the properties ofrecursively enumerable languagesthat very easily we can say whether it is a decidable or undecidable. We need not have to go through the entire proof. Once we know that it isa non-trivial property immediately you can say it is undecidable, but what about the properties of Turing machine? What is what do you mean by aproperty of a Turing machine? Certain properties like there is a Turing machine have five statesorsimple things it isa decidable.There are other propertieswhich are undecidable. For example, given a Turing machine start it on a blank tape will it ever print three 1’s consecutive 1’s.

Without loss of generality assume thatthe alphabet isthis. Then start on a blank tape,will it ever print three 1’s? That is an undecidable problem. So, properties about Turing machine some of them are decidable, some of them are undecidableand there is no systematic way of approaching is that. Each problem you have to look into the problem and find out whether it isundecidable or not. So,the question is;Does a Turing machinestartedon a blank tapeever printthree or not even on a;does it(No audio from 47:20 to 47:27)three 1’s three successive1’s. This is undecidable,how would youprovethat this is undecidable? GivenM,constructM capwhichsimulatesM onblank tapeblank tape.(No audio from 48:09 to 48:16)

Now, forMfor M the symbols are 0, 1. For M cap0 is represented as 0 1; 1 is represented as 1 0.(No audio from 48:27 to 48:34)So, suppose inM, I have in M,I have at, but at some stage I have the tape0 1 0 0. M capI will have the tape0 is represented as 01;1 is represented as 100 101. So, whatever tape is here in M, non blank portion of the tape is there forM.M cap if we represent it will not have three consecutive 1’s because 0 is represented as 01;1 is represented as 10and also when you make 0 into 1,01 has to be made into 1 0. And you do it in such a way that first you change the 1 to 0 and then the 0 to 1. One by one only you have to change. If you change 0 to 1 first at one stage you will have two 1’s.

Now, that you have y if you want to change 01 to 10 first, change the 1 to 0and then the 0 to 1. Similarly, when you change,you want to change 1 to 0;1 0 has to be changed to 01.So, first change the 1 to 0 then the 0 to 1. This way you make sure that always 0you get a change the 1 to 0first avoiding at any stage appearing three 1’s appearing. So,M bar is just M where every 0 is represented as 01 and 1 represented as 10 and because of the way we have to chosen the representation for 0 and 1. It will not have three consecutive 1’s.

(Refer Slide Time: 50:49)



IfM accepts,M bar M cap will print three 1(s). So, if the problem whether Mmachine will print three consecutive 1(s)or not is decidable. You can start M on blank tape, and see whether it will accept or not. What is that problem? The problem isdoesM haltin anacceptingstatestartedon blank tape,and this is equivalent to the problemdoes epsilon belong to L of M?Does epsilon belong to L of M or L M contains epsilon. That is a non-trivial property, isnot it? It is a non-trivial property does this epsilon belongs to L of M is a non-trivial property,by rice's theorem that is undecidable. Now, if you assume that the problem whether the Turing machine will print three 1(s) at any time is desirable,then you come to the conclusion that does epsilon belong to L of M is decidable, but that is an undecidable problem.

So, does a Turing machine ever print three 1, this is undecidable. Now, this is a property of the Turing machine,it is not a property of the recursively enumerable language. It is a property of the Turing machine. Similarly, there are several problems about Turingmachine,there is no systematic way of tackling that. Rice's theorem helps us to tackle the problem decidability or undecidability of problems on recursively enumerable of languages, but aboutTuring machines,each problem you have to look at it, and find out a solution. So, this is something aboutdecidable, and undecidable problems of Turing machines and recursively enumerable languages. Next, we have to consider some moreproblems like post correspondence problem, and using that how some decidable problems like the ambiguity problem can be solved. So, we shall consider in the next class.