Theory of Computation

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

Lecture No. #30

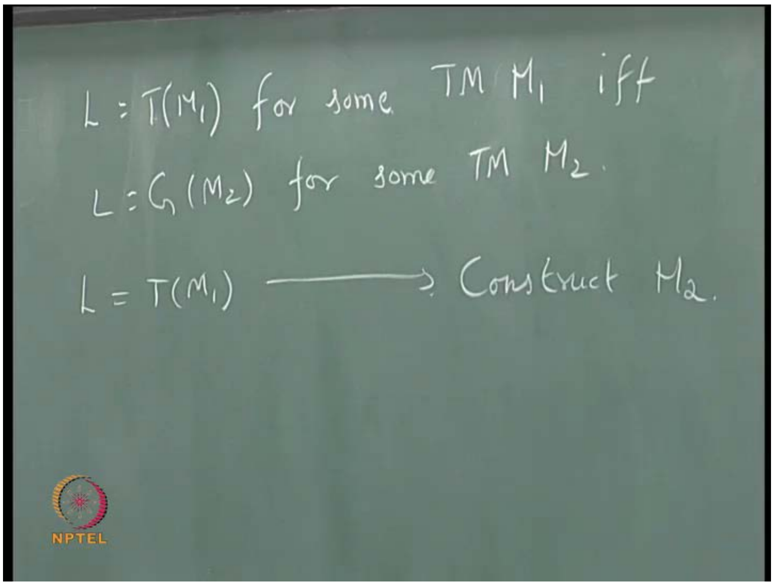
Turing Machine as a Generating Device

(Refer Slide Time: 00:17)

So, today we shall consider the Turing machine as a generating device.We have consideredTuring machine as an input output device computing somethingand we have also looked at the Turing machine asan accepting device, accepting type zero languages.We can have a Turing machine, where there is an output tapeother tapes are there, but there is an output tape, theTuring machineand in this output tapestrings are printedwithin hash marks,some strings are printed on this tape and this tape head always moves rightand something written once is not erased and it is blank towards the right.First it writes a hash symbol, then some string, then hash symbol,then another string hash symbol and so on. So, whatever is appearing between two hash symbols, that string is set to be generated by the Turing machine.If this Turing machine is denoted as M,then G of M,we willusually denote acceptance by T of M or L of M usually by T of M.The language generated by this Turing machine using an output tapedenoted as G of M iswwisgeneratedby M.Whatthis means isthis w will appear between two hash symbols in the output tape,the set of all strings is denoted by G of M.

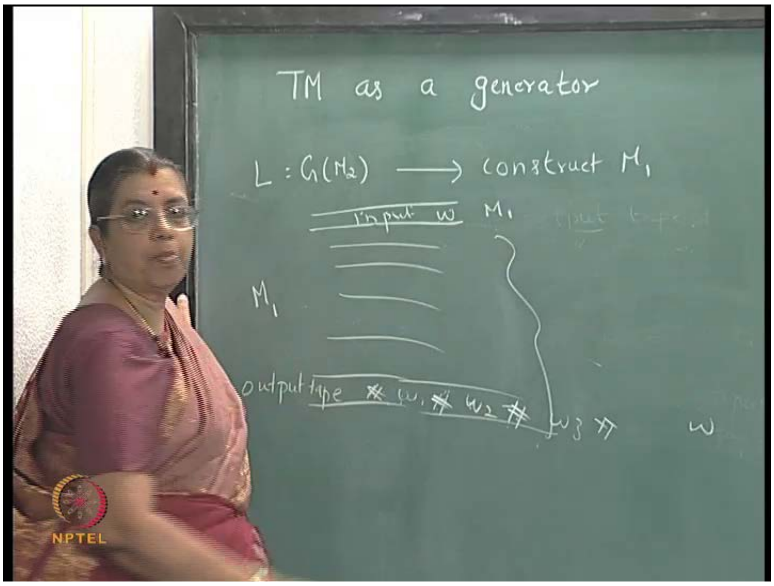
Now, there is no necessity that the string should be generated any particular order, we know what is meant byStandard ordering on strings,Lexicographic ordering on strings, all those things we know.It need not be in any particular order and it is also not necessary that one string should be generated only once, the w 1 is generated here after sometime it may again be generated,there is no such restriction.Strings are generated between hash symbolsand all such things belong to G M and it is not necessary they should be generated in any particular order.Longer strings can be generated first shorter strings later and so on.And also one string may be generated many timesit does not matter, but if the strings are generated in the standard ordering manner lengthwise increasing and within the same length in the lexicographic ordering,then that language is a Recursive set.What is a Recursive set?A recursive is a set accepted by Turing machine which halts on all inputs.

(Refer Slide Time: 03:53)



So, weget this two characterizations now. So, the first thing is we want to show that if L is equal toTof M 1 for some T M, M 1,if and only if L is equal to G of M 2 forsomeTuring machine M 2.That is if L is accepted by a Turing machine M 1 then L is generated by some other Turing machine M 2 and vice versa. So, we have to prove intwo directions. So, L is equal to T of M 1 then construct M 2,thenother way around L is equal to G of M 1 then constructM . So, the second portion will take first this is1.

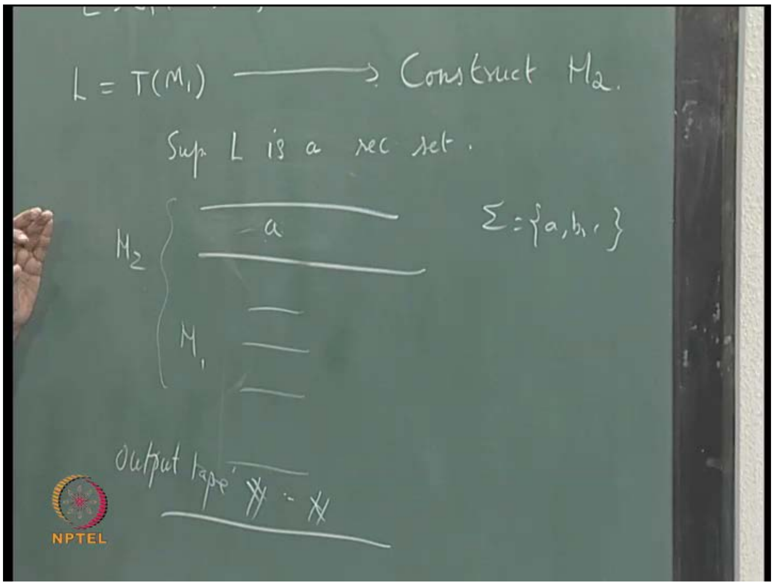
(Refer Slide Time: 05:25)



So, second is L is equal to G of M 2 thenconstruct M 1,now how do you go about this you have a Turing machine it can have many tapes, but one output tapewhere strings will beprinted.Now, how do you construct M 1 which will accept the same language M 2 has all this M 1will haveone more input tapeand input is kept here, M 1 has all this plus theinput tape, M 1 simulates M 2 andit generates one string.When up to the point where one hash symbol appears M 1 stimulates M 2,then when a hash symbol appears M 1 comparesthis string generated in the end, last string generated with the input,if they are the same it will accept, if they are different it will proceed further, if they are the same it will accept; otherwise it will generate the next symboland then up to the point it generatethe hash symbol and it will simulate M 2.

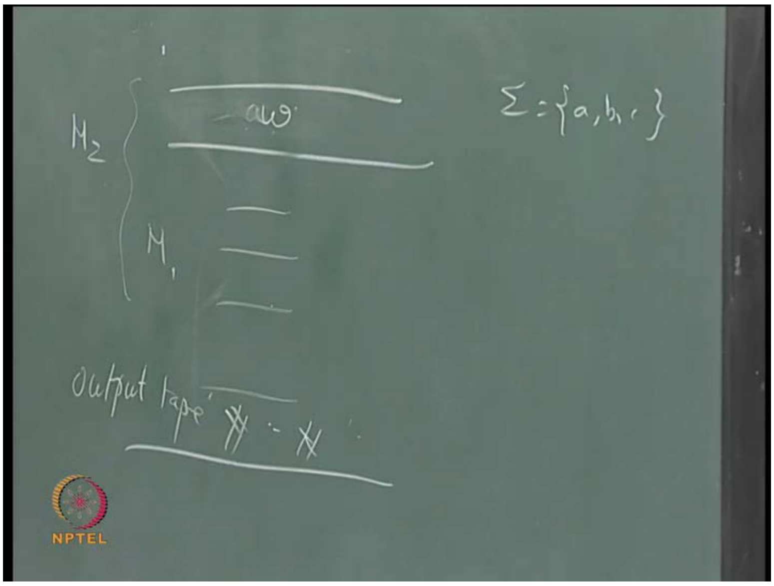
Once this generate the hash symbol it will switch back and compare this and thisif they are the same it will accept, if they arenot the same it will proceed furthergenerating.If at some stage input is wif w is generatedthen at that stage the comparison will sayand it will accept.So, if L isgenerated by a Turing machine M 2,then you can construct a machine M 1 which will accept the same language.

(Refer Slide Time: 07:48)



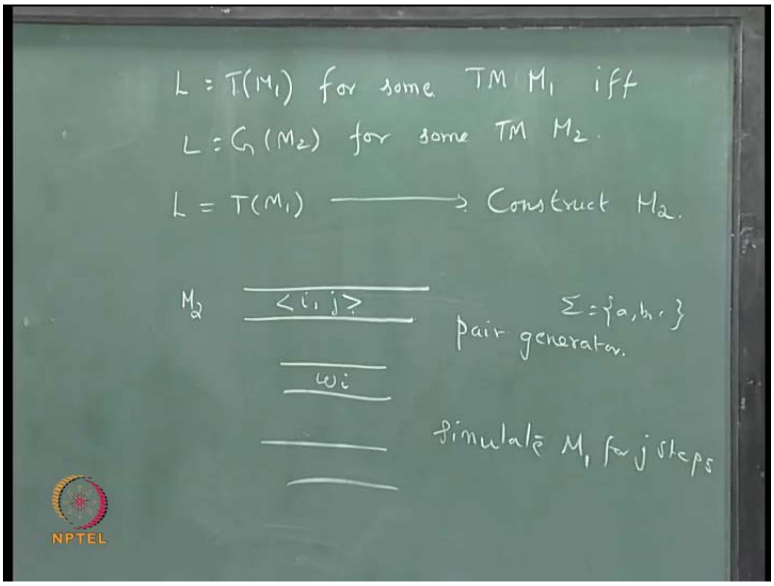
Now the other way around if L is accepted by a Turing machine how will you construct M 2?Ifthe Turing machine halts on all input one by one we can take this strings,suppose the Turing machine halts on all inputsit is Recursive, then one by onein the standard ordering you generate the strings,suppose alphabet a this a b c a thensuppose L is a Recursive set.Then the Turing machine willhalt on all input, one by one you generate the strings, the strings can be any minute.Firstone suppose the alphabet is a b cthen a will be generated first,you keep one tape as output tape, M 2 will simulate M 1 and if this is accepted it will print here,if it is not accepted it will take the next symbol try that if it is accepted it will print like that you can operate.

(Refer Slide Time: 09:41)



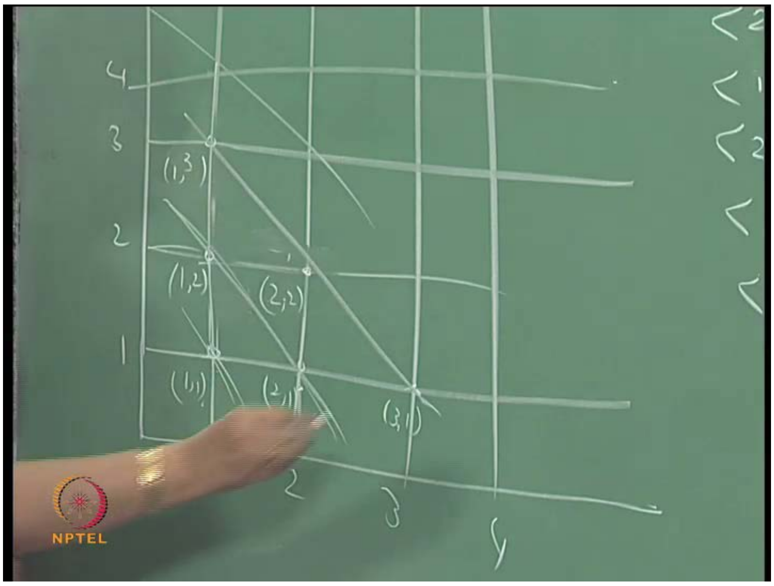
But the Turing machine need not halt always this is not truein that casewhat will happen?What happens is see if there is a string wit will simulate M 1 and if it accepts it can print, but if it does not accept it will gets into a loop what happens?It cannot keep on simulating the behavior of M 1 on w infinitely,then it will not be able to do anything.See the Turing machine can reject an input by halting inanon final state or by getting into you loop,suppose it gainsstraight the loop this also we will get into loop and you will not be able to proceed further.If it is rejected you must go to the nextone and so on. So, how do you know that? So, you cannot simulate the behavior of M 1 on any string infinitely,if you start doing that you will be in trouble.

(Refer Slide Time: 10:44)



So, whatthis machine M 2 doesis M 2 has one tapewhere hashs of integers i, j aregenerated,pairs of integerswhatever notation we can take decimal,binary i j you can take, it generates pairs of integers.How canyou generate one by onepairs?Unless it is countable you cannot generate there should be some orderright.

(Refer Slide Time: 11:27)



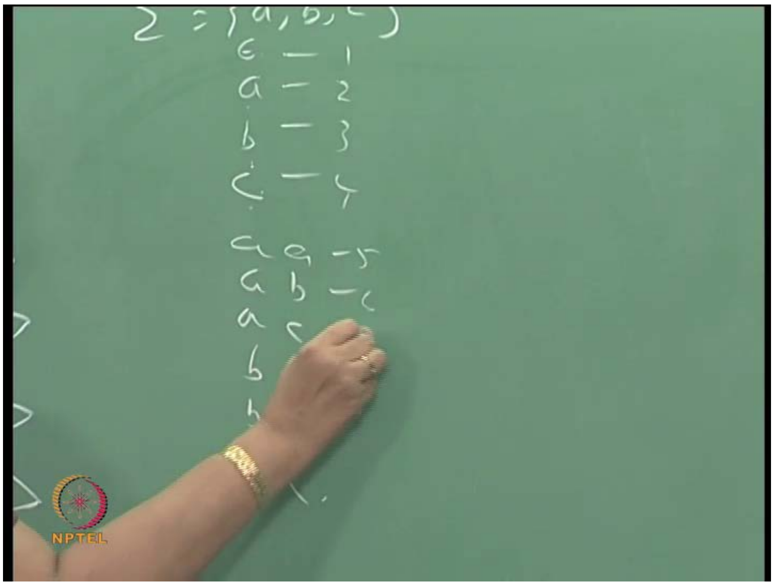
So, how can youhave the pairs generated?Suppose I have thisinteger grid pointsthe X and y coordinatesrepresent pairs of integers. So, this is 1 1and this is 1 2,what is this?Let me draw the grid againwrite this here. So, this is1 2,3 4, 1 2,3 4 and so on. So, this is 1

1thenthis is 2 2 this is 1 2this is 2 1 this is1 3, 2 2, 3 1 and so onyou take theseinteger points. So, if you want to have them as a countable setthis is counted first, then all points in this diagonal, then all points in this diagonal and so on. So, the set of pairs of integers is countable. So, which one will come first 1, 1will come first,the some should add up to 2 now, then some should add up to 3 1, 22, 1 3 then 4 3,1 3, 2 2, 3 1 and so on.

So, that is this point, then this 1 1,1 2,2 1,then in this diagonal 1 3, 2 2, 3 1,the next diagonal will be 1 4, 2 3, 3 2, 4 1 and soon. So, the pairs of integers is countable one by one you can generate them.

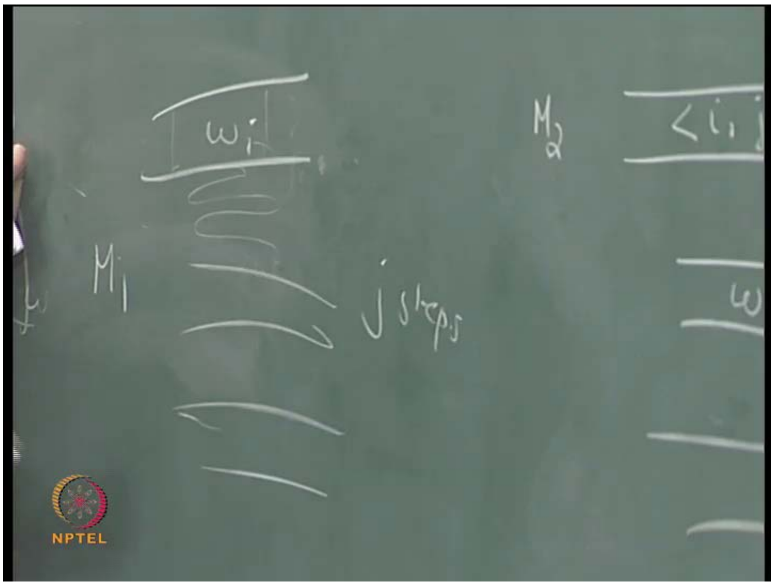
So, in one tape we generate the pairs of integers i j, i varies from one to infinitely, j varies from one to infinity,after generating this have a tapewhere you generate the i-th string in theenumerationover any alphabet and you can enumerate the strings using standard i ordering.

(Refer Slide Time: 14:19)



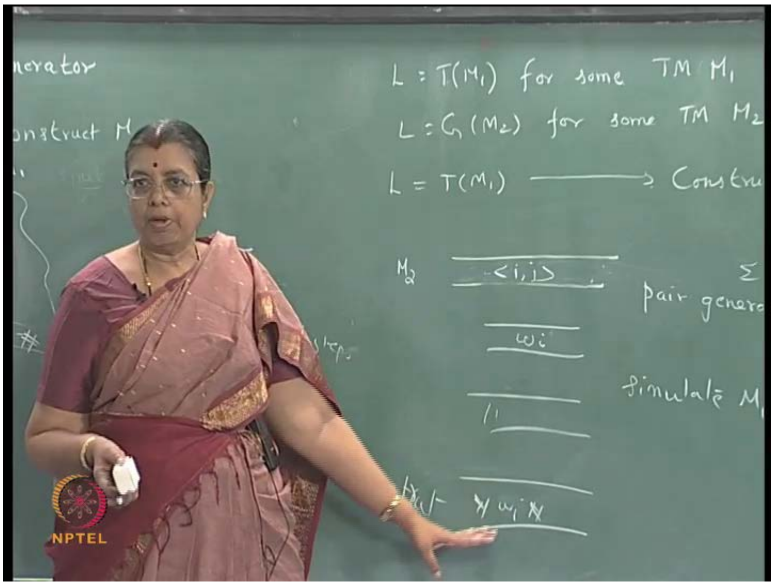
Suppose I take a b c, sigma is equal to a b c how can you enumerate the strings?If you want to include, include epsilon alsoa b c,aaa b,a c b a,b bb cand this is first string means,again thirdfourth fifth sixth seventh. So, seventh string would be in a c and so on.So, when the pair i j is generatedyou generate the i-th string in the enumeration, thensimulate M 1 forj steps.

(Refer Slide Time: 15:10)



If M 1 accepts what is the behavior of M 1,suppose M 1 accepts w i,then it will have it on the input tape that may be other tapes.Without loss of generality you can take it as a multi tape, similar tapes should be there here also.Then it will simulate a M 1 will move on w y and after some steps j will acceptright after some steps, if it does not acceptit may halt inanon final state or it may get into loop.Now, what you do is you generate the pairs one by one and when the pair i j is there you generate the string w i,you knowwhich is the string the i-th string in theenumeration and you can generate that.

(Refer Slide Time: 16:29)



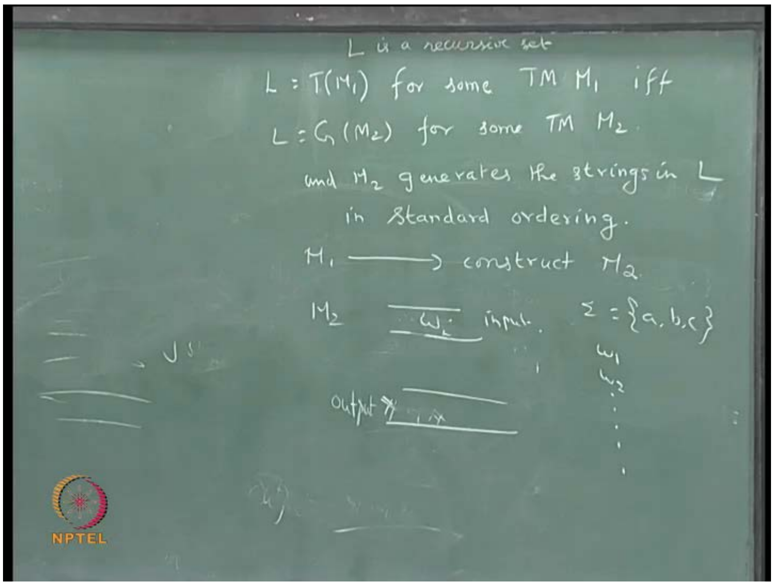
Then similar to M 1 for j steps if M 1 accepts on the j-th step not before or after on the j-th step if it accepts,if this has an output tapeyou generate w i if it does not accept on the j-th step then go to thenext pairthat may be some i dash j dash.Then generate w i dash, then simulate M 1 for j dash steps. If it accepts on the j dash steps you generate it on the output tape,if it does not go back to the next pair in the pair generate. So, if at all a string w i is accepted, it will be accepted by M 1 in somej stepsand at some stage or the other the pairi j will appearon the pair generated,i j will appearhere at some stage or the because one by one you are going to generateand when the pair i j is generated w i j will be generated here and the machine simulating M 2,when simulating M 1 will accept in j steps on the j-th step, so it will be printed on the output.

So, if it is accepted byM 1 it will be generated, if it is not accepted no stage you will get that i j pair,either it will get in to a loop in which case w in n plus one all those may appear, but it will never get accepted or if it is halts ina non acceptingstage.Also it will not get accepted you will not get that correspondingi j pair. So, if a string is accepted by M 1 at some stage or the other it will be generated by the machine M 2 between two hash symbolsrightand if it is not accepted it will not be generated.

So, if L is equal to T M then you can construct the machine M 2 in this manner right. So, acceptance and generation in a way when onemachine can accept another machine can generate the same language, because these strings can be generated in some order that is why it is called Recursively enumerable set.The name came because it can be generated by the Turing machine,not only that in this construction you will note that one string will be generated only once, of course, there is no necessity in the definition that one string should be generated only once, but in this construction you will see that one string will be generated only once.The reason isyou generate that string on the output tape only when the pair i jappearsand w i is generated and it is accepted by M 1 on the j-th step.Suppose i j plus 1 appears,then w i will be generated it will simulate for j plus 1steps, but on the j-th step itself itis accepted.At that time again it will generate w i, it will generate only when the machine M 1 accepts when w i is accepted on the j-th step.

So, one string will be generated only once in this construction,because in thegeneral definition we are not putting that restriction.

(Refer Slide Time: 20:17)

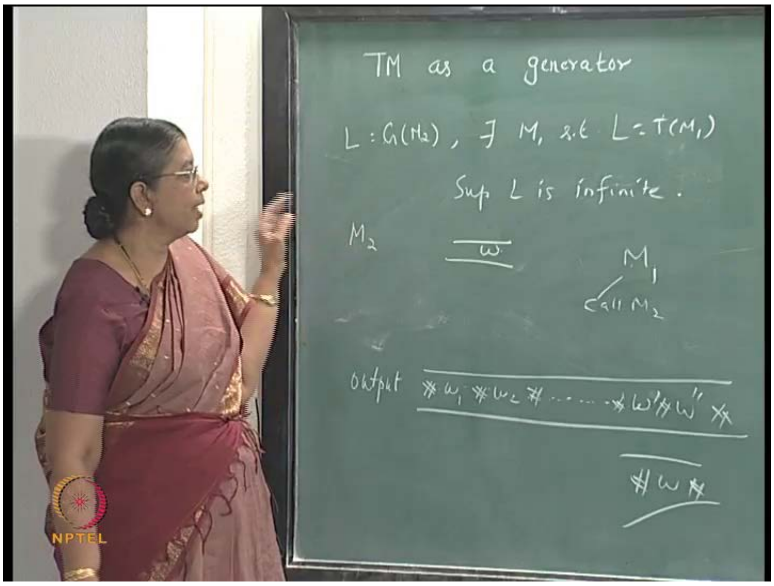


Now, inthis termwe can also havea nice characterization forrecursive sets.Suppose L is a recursive set,then L is equal to T of M 1 for some T MM 1,that means, this M 1 will halt on all inputs, what is the condition if L is a recursive set?It will halt on all inputs.If and only if L isG of M 2 for theT MM 2 and M 2 generatesthe stringsL instandard ordering.Now given M 1 construct M 2 and vice versa given M 2 construct M 1,now given M 1 how will you construct M 2?Supposethe alphabet is a b cone by one it will takefirstthe a will be taken, then b, then c as and a a, a b, a c and sooneach string will be taken.

So, first w1 will be takenand then w 1 is a of course, then it will simulate the behaviorof M 1, M 2.We willwrite the first1 on the input and then simulate M 1,see M 1 will always halt, either it will accept or reject it will always halt, when it halts if it acceptsin the output tapeyou printw 1if it halts, but does not accept leave itgo to the nexti,write the next one in the enumeration, simulate M 1 with this ultimately it has to halt, because it is a Recursive set.When it halts if it accepts write it between two hash symbols, if it does notaccept leave it and go to the next string in theEnumeration is that clear.

So, in thisinput tapein one tapeyou will generatew 1, w 2,w 3 etcetera one by one,when w i is generatedit will simulate the behavior of M 1, M 1 will always halt. So, either it will accept of not, if it accepts it will be printed on the output tape, but does not accept go to the next w i plus 1 right. So, this is the way this works.

(Refer Slide Time: 23:53)

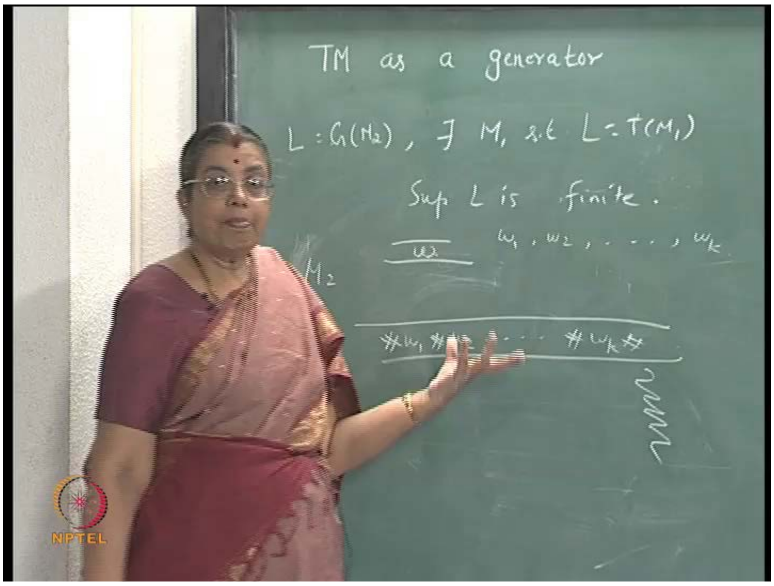


Thenin this caseif L is equal to M 1, L is equal to G of M 2 there exits M 1 such that L is equal to T of M 1,here in one case we will not be able to really construct the machine.We only prove that there will be a machine M 1 such that L can be accepted by the Turing machine.There is a subtleT hereif L is infinite no problem, the problem arises in a setspecial case when L is finite. So, suppose L is infinite then in this case no problem.How does M 2 behave? M 2 has an output tapewhere the strings will be printedin standard orderingright?

So, how does M 1 behavegiven a string whow will M 1 accept it or not?What M 1 does is it will call M 2 as a sub routine andkeep on printing the strings here,at some stage w will be printed,this is in a Canonical order length wise and within the same length lexicographic ordering. So, at some stage w will be printed between two hashesor some string w dash,which is less than wwill be printed next w 2 dashwhich is greater than w will be printed.If w is notin the G M 2 then some string w dash which occurs beforew in the enumeration will be printedand next w 2 dash which occurs after w in the enumeration will be printed.If w is in G M 2 it will be print, sotwo possibilitiesthe either w will beprintedor some w 2 dashwhich occurs after w in the in the enumeration will be printed.

So,M 1 keeps on checking whether w is printed or not,if w is printed it will halt and accept,if w is not printed after some w dash, w 2 dash is printed whichcomes after w in the n; that means, w is not generated by theTuring machineM 2 . So, in this case it will halt and reject,either way M 1 will halt on the inputw and say yesor no, this is then L is in infinite there is no problem, but if suppose L is finite there isa small problem here.

(Refer Slide Time: 27:18)



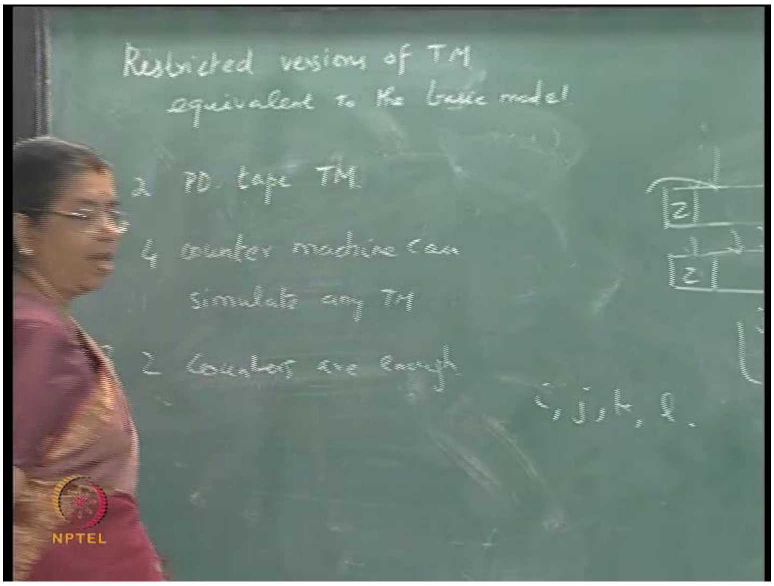
What may happen is M 2 can it is say w 1 has gotk strings w 1,w 2,w k,finite set it has got only k strings. So, M 2 can print one by one,now after printing w k,we are only saying that it will print all the strings in G of M 2,there is no necessity for M 2 to halt after printing the last respond.If it halts well and good, but there is no necessity that M 2 should halt after printing w k,what it may do is after printing all these symbolsit may get into a loop.It does not do anything it isjust printed the k strings afterwards it just gets into a loop.

So, whatever is printed on the tape between two hash symbols is G of M 2 . So, this G of M 2 consists of k string w 1,w 2,w k,in this case that construction which we have just discussed will not work,because given some string say w is greater than w k in the enumeration w occurs, after w k in the enumeration,then neither w will be printed here not something greater than w will be printed here, after printing w it gets into a loop. So, in this caseif you follow that construction M 1 will also have to going to a loop, but if it is a recursive set M 1 has to halt really, it has tohalt whether it accepts or not.If itgoes to a final state it will accept, if it goes toanon final state will reject, but it has to halt, but this sort of a simulation which we considered for the infinite set will not work for the finite k,if at the end M 2 gets into a loop.

So, in this case we may not be really able to construct the Turing machine, but this happens only when G of M 2 is a finite set, but any finite set can be accepted by a finite state Automaton and when it can accepted by a finite state Automaton it can definitely be accepted by a Turing machineright? So, we can construct a Turing machine which will accept just G of M 2 even if it is finite, infinite we know the construction,finite we know that we can construct a Turing machine, because we can always construct a finite state Automaton for this, but given M 2 whether G of M 2 is finite or infinite we do not know,if it is infinite we will be able to construct the Turing machine, if it is finite weknow that a machine exists, but we will not be able, we cannot say thatand given a M 2 whether G of M 2 is finite or infinite is an undesirable problem.

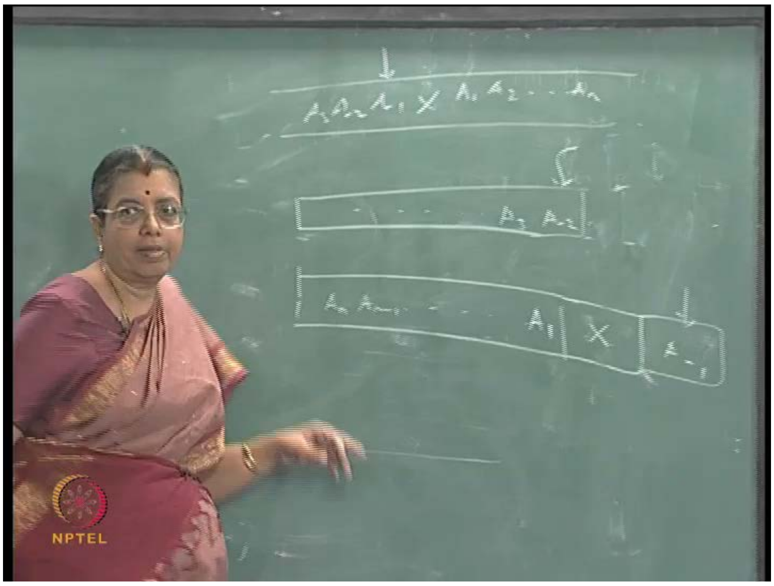
So, that we cannot say here after we will come across the setsorthis sort of an argument,which we will next class also we shall start. So, we do not know whether G of M 2 is finite or infinite?If it is infinite we know how to construct the Turing machine M 1 which will acceptthe same thing, halting on the inputs.If it is finite we know that a finite state Automaton will always exits for thatand once it can be accepted by a finite state Automaton it can definitely be accepted by a Turing machine right? So, thisthere is a subtle argument there we have to take of thatin the proof. So, this is. So, for ofas considering the Turing machine as a generatorand we have considered generalized versions of Turing machine,which are equivalent to the basic model.Now, we shall consider restricted versions of Turing machine which are equivalent to the basic model.

(Refer Slide Time: 32:06)



Restrictedversionsof Turing machineequivalentto the basic modelone istwo Pushdown tape,Turing machine has two tapesand both of them behave like a stack,only the top most symbolcan bereadand you can push symbolspop symbols and soon.

(Refer Slide Time: 33:14)



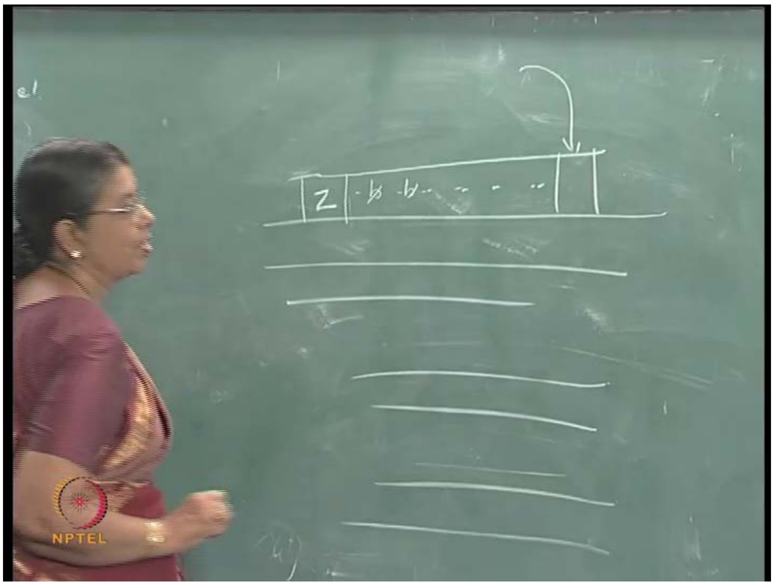
So, the Turing machine has two tapesand this istop of the stackand again another one which is thetop,with this we can simulate any Turing machine two tapes it has got they behave likestack,stack means you can increase the stack.

So, as you can print more and more symbols here by pushing, soas many symbols youwant you can addhow can you do this? So, if you have the Turing machineand at some stage it containsA 0and A 1, A 2,A n,A minus 1, A minus 2, A minus 3 on the tape,then there is a current head position.In this one it will be represented as A n,A nminus 1,A 0,A minus 1, A minus 2, A minus 3 and soon.All this again tape contains in the top of thecurrent symbol read by the Turing machine, suppose the next move it printsX andmoves right. So, what will happen isthis willprintX and thenthis will be transferred hereI am sorryit goes to A X will be A 1, X will be printed here,this tape head will move rightto here.

If it moves left in that case X is here this becomethis head will be here. So, whatever manipulation isoccurring only occurring at the top, X is pushed hereandthisremains as it is.Now suppose the move is to the leftoriginally you had A 0 here,supposeit is moved here this is change into X and then this is pushdown,this tape head will move, A 1 will be popped outfrom that,A minus 1 will be popped outfrom here and it will be add pushed here.

So, again the current symbol read is A minus 1just this. So, any Turing machine with one tape we can simulate by two pushdown tape,of course a multi tape Turing machine can be simulated by A 1 tape Turing machine. So, any Turing machine you can simulate with a just two pushdown tape,this is a restricted version it has got two tapes, but both of them are behaving like a stack,stack is very restricted compared to the general Turing machine tape.Then general Turing machine tape we can move anywhere on the tape, can print symbols anywhere you want and soon, but stack manipulation will be done only on the top, either you can push of pop and soon. So, any Turing machine you can simulate withtwo pushdown tapes,this is a restricted version equivalent to thebasic model,then another is4 counter machinecansimulateanyTuring machine,what is a counter machine?counter itself tells you thatit keeps a count.

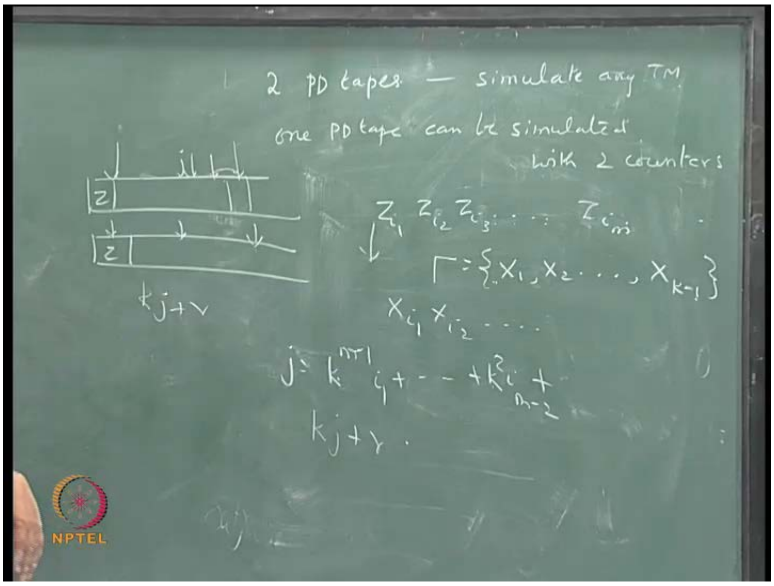
(Refer Slide Time: 37:44)



So, the Turing machine will havesometapeswhere there will be one symboland rest of them will be blank,there will be one non blank symbol here and rest of them will be just blank.The tape head will be somewhere herethis is called a Counter, it does not write anything on this tape,the number of cells from Z the tape head is in this cell,suppose it is atthe tenth cell the count is ten. So, you count from this Z till the head position that isanumber isnot it? So, this tape tells you that number. So, this tape gives you only the information ofa number. So, we can show thatlike that if we have 4 counters they are very restricted versions ofa tape isnot it? It has got onlyone non blank symbol rest of them is blank, you can move left and right does not matter, but you do not print anything, tape head will just move on the tape.

And the only information you have is the number of cells from the non blank symbol,there is a head from the non blank symbolit just keeps a count,such a tape is called a Counter.So, with 4 such tapes you can simulate any Turing machine,if you have 4 suchtapes which behave as counters we can simulate any Turing machine and how can you do that?The way you do is that is like this.

(Refer Slide Time: 39:57)



You know thatwith 2Pushdown tapesyou can simulateany Turing machine,that is what we have just see simulateany Turing machine,now if you show that 1Pushdown tapecan be simulatedwith2Counters,we show that1Pushdown tape can be simulated by 2Counters,2Pushdown tapes can be simulated by 4 counters.Now how can you simulate 1Pushdown tape with 2Counters?the pushdown tape will have something like Z M, Z 1something like this isnot it,write it as a string Z 1,Z 2,Z 3,Z M,this is the content of the Pushdowntape right?These symbols come from an alphabet.

So, the alphabet Gammahas say X 1,X 2,X k, k minus 1 symbol including theblank symbol it has got k symbols. So, this Z 1is one of theXsright? Z 2is again one of thek minus 1 symbol. So,I would write instead of 1 2I will write it as i M which will be more clear Z i 1, Z i 2, Z i 3,i M,each of them will bea symbol from here. So, i 1 will be say 5,i 2 will be 7 something like that isnot it? So, I mean actually it is the same symbol. So, Iwill write it as X i 1, X i2, X i m.

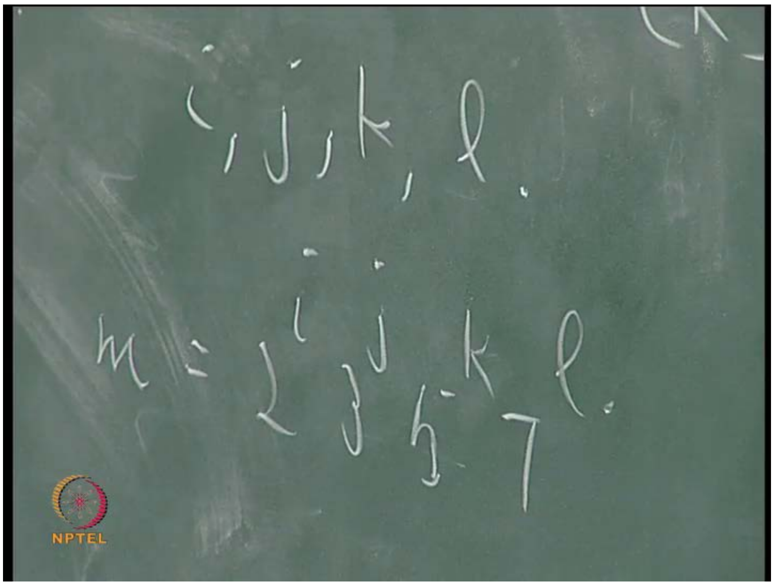
Now, you lookat an integer j which isi M timesplus k times i M minus 1 plus k square term i M minus 2 and soon,k power M minus1timesi 1,k having M minus one symbols here, soyou take the base kand write this.Now from j will you be able to get i 1,i 2,i 3,etcetera,you can uniquely get back the string.Divide j by k whatever reminder you get that will tell you what is i M out of the last symbol.Take the quotient divide by k that will tell you the reminder will tell you what isi M minus 1 right?again take the quotient divide by k that will tell you what is this.So, this is the content of the pushdown storecan be simulated like this,suppose Iadd X rhere I am pushing one more symbol how will j change, j will change tok j plus risnot it. So, in 1 counter Ihavejtheninstead of thisImust have k j plus r right?Iwill try to have that in this tape the alternativelyIcan use them.

So, first Ihave j here Iwant to have k j plus r here, how will you do that?This head will be here initially, move this cell 1 to the left,move this 1k cell to the right, movethis 1cell to the left, move this 1 k cell to the right,when this reaches herethis will be somewhere here representingk j,then move it or more cells, then you will havek j plus r in the second tape right?When youwant to removepopping or when you pop what happens?you must havein 1 counter you are having j,in this counter you must havej by kthe floor of that right?

How can youget ithere?The tape head is pointing heremovek cells to the left,move this 1 cell to the right,move this k cell to the left, move this 1 cell to the right .So,finally, when it tries to move left of this Z you stop, at that time you will havethe quotient when j divided by k in this second tape. So, counter is a specific kind of a tape where there is only 1 non blank symbol and the head position tells you how many cells it is from the non blank symbol, that is it gives just an integer, an integer can be stored in a count nothing else.And making use of that you can show that any Turing machine can be simulated by 4Counters the method is you know that any Turing machine can be simulated by 2Pushdown tapesand we show that 1Pushdown tape can be simulated by 2Counters.

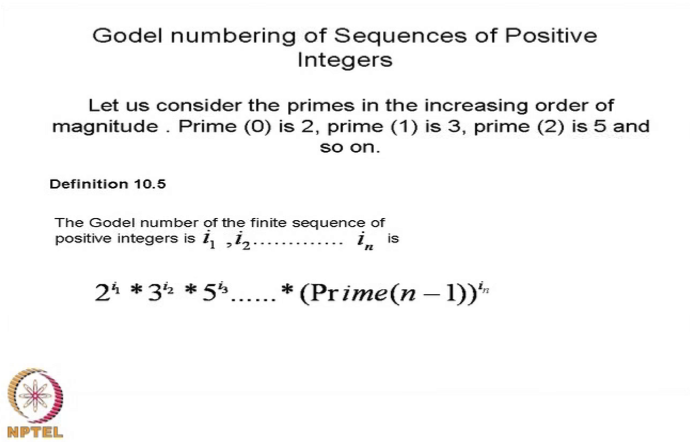
Now, anyTuring machine we have seen thatcan be simulated by 4Counters, it can also be simulated by 2Counters, any Turing machine we can simulate with just 2 counters, how can you do that? So, 4Counters we can show that you can simulate with 2Counters.

(Refer Slide Time: 47:37)



So, actually the 4Counters,2Counters are enough,actually 4Counters you store four valuesi j kl,we can store it by1 number M, 2 power I,3 power j, 5 power k,7 powerl,from this we can retrieve back the values ofi j klright. So, in one counter you can store this, the other counter isfor manipulation, when you want to increase i by 1 you have to multiply it by 2, when you want to increase j by 1 you have to multiply by 3 and so on for thatthe other counter will be usedthis idea is called Godel numbering right?So, just a little bit about moreon restricted versions, then we will go to universal Turing machine and the Halting problem.

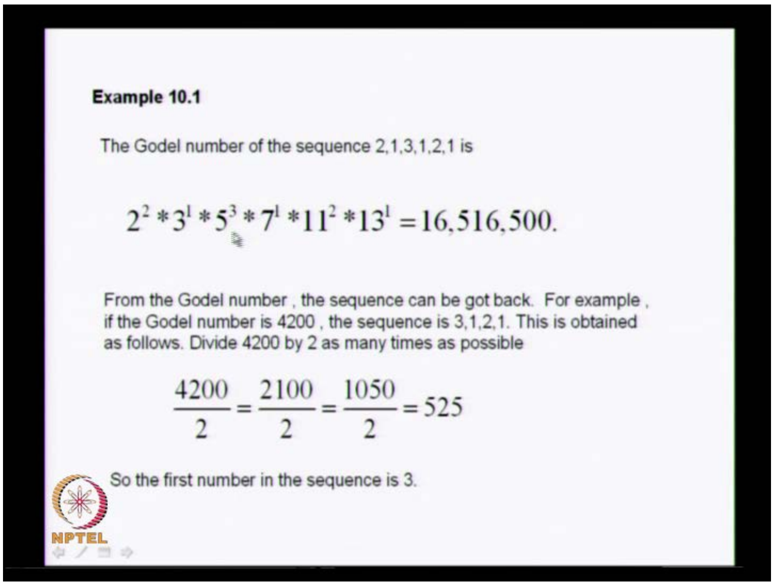
(Refer Slide Time: 48:37)



We have used the idea of Godel numbering in constructing the Counter machine, let us see what is a Godel numbering this is an important concept which is the basis of Godels incompleteness theorem, which you know is a very famous theorem.Godels incompletenesstheorem says that there is no complete in consistent system for integer arithmetic or any axiomatic system per integer arithmetic has statement which areproved, but which cannot be proved from the axiomsfor that he used an idea which is called Godel numbering.We are also using that idea in the construction of counter machines.Now let us see what is a Godel numbering, first of all let us consider the Godel numbering of a sequence of positive integers.

So, now let us consider the primes in the increasing order of magnitude. So, a prime 0 is 2 this is a first prime,we call 3 as prime 1,then prime 2 is 5 andso on the Godel number of the finite sequence of positive integers i 1,i 2,i n, is 2 power i 1, 3 power i 2,5 power i 3 and prime n minus 1 power i n.

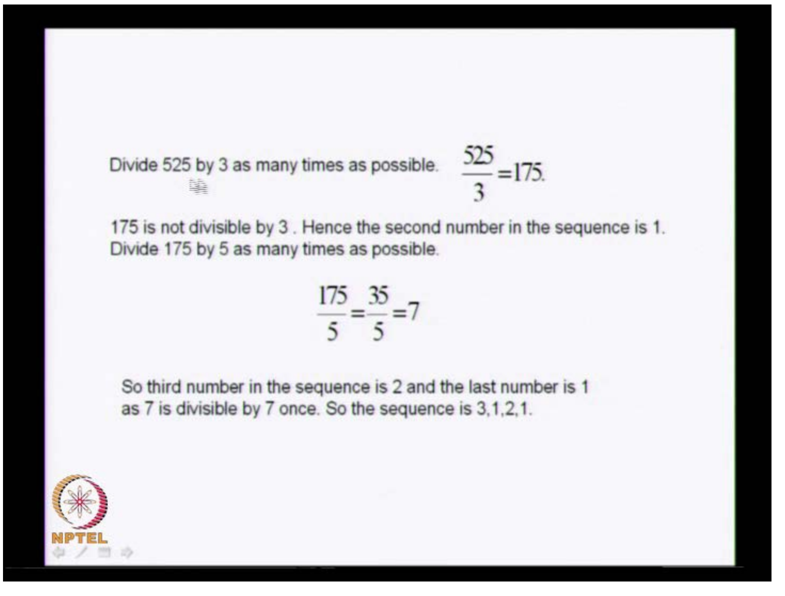
(Refer Slide Time: 50:08)



For example, suppose we have the sequence 2, 1, 3, 1, 2,1,then the Godel number of this sequence will be 2 power 2 because this is 2,then 3 power 1 second number is 1,next prime is 5, sothenext number in the sequence is 3. So, you have 5 power 3,then 7 power 1,next prime is 11 and the next number here is 2, so11 power2, 13 power 1 and if you calculate this value it is a very large number.And given a sequence you can find the Godel number of that sequence which is a verylarge numberand given a Godelnumber this sequence can be obtained in a unique mannerfrom the Godel number the sequence can be got back.

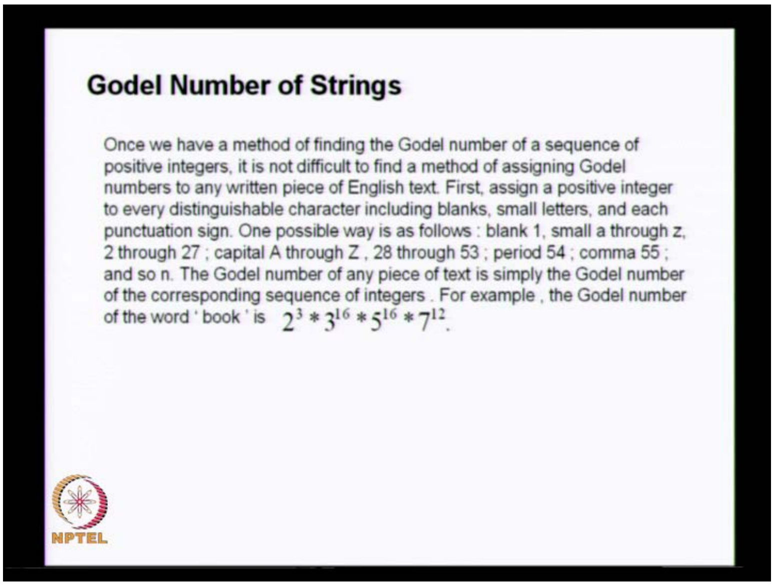
Now, for example, suppose the Godel number is 4200 the sequence is 3,1, 2,1how do you get this?This is obtained as follows, divide 4200 by 2 as many times as possible. So, 2 divides 4200 3 times and you get the quotient like this. So, the first number in the sequence is 3.

(Refer Slide Time: 51:41)



Then you try to divide525by the next prime which is 3it divides only 1. So, the next number in the sequence is 1, the quotient is 175 nowand try to divide by 5 and it divides 175 2 times and you end up with7 as the quotient. So, the 3 number in the sequence is 2 and the last number is 1,because next prime is 7 7 divides 7 once. So, the sequence is 3,1, 2 1. So, given a sequence we can find the Godel number and given the Godel number we can find the sequence.

(Refer Slide Time: 52:26)



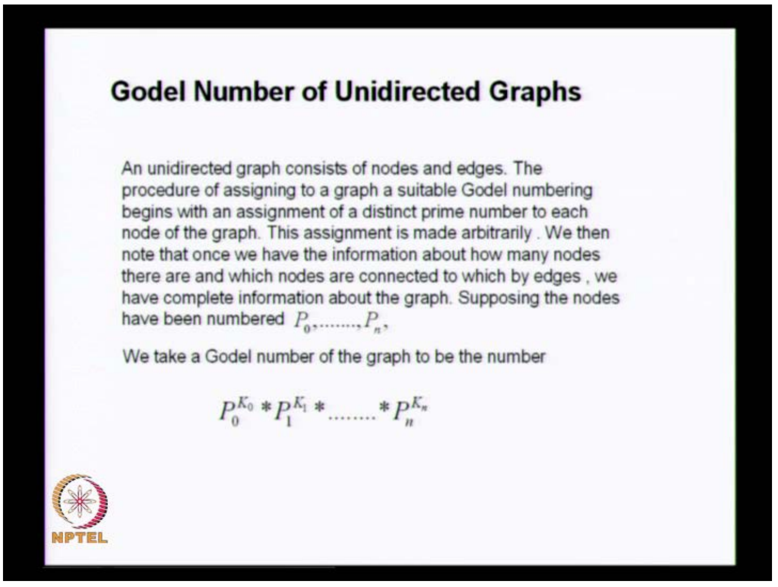
Now, the same idea can be extended to strings and this is theidea used by Godel in his theorem.Once we have method of finding the Godel number of a sequence of positive integers, we can extend it to strings.How do we do this?Any written piece of Englishtext you can assign a Godel number.Now assign a positive integer to every distinguishable character including the blanks, small letters of the Englishalphabetand punctuation signs,one possible way of doing this will be blank you denote by1,small a to small zyou denote by numbers 2 to 27,small a stands for 2and 3 stands for small b,4 stands for small c and so on,then capital A through capital Z use the numbers 28 to 53. So, capital A will be 28 capital B will be 29 and so on, then full stop we can use number 54 comma use the number 55 and so on.

The Godel number of any piece of text is simply the Godel number of the corresponding sequenceof integers for example, the Godel number of the word bookis this why b is 3,o is 16,o is again 16 and K is 12.

So, the sequence ofintegers will be 3, 16, 16, 12 and choosing the corresponding primes you get 2 power 3 into 3 power 16 into 5 power 16 into 7 power 12,this is the Godel number of the word book,So, any English sentence you can have a Godel number, you must remember that the Godel number will be a very a large number.What Godel did in his incompleteness theorem is, he made a statement and the Godel number of that statement he calculated. So, the statement was something like this the theorem whose GodelnumberX is not a theoremand when you calculated that number it was X.

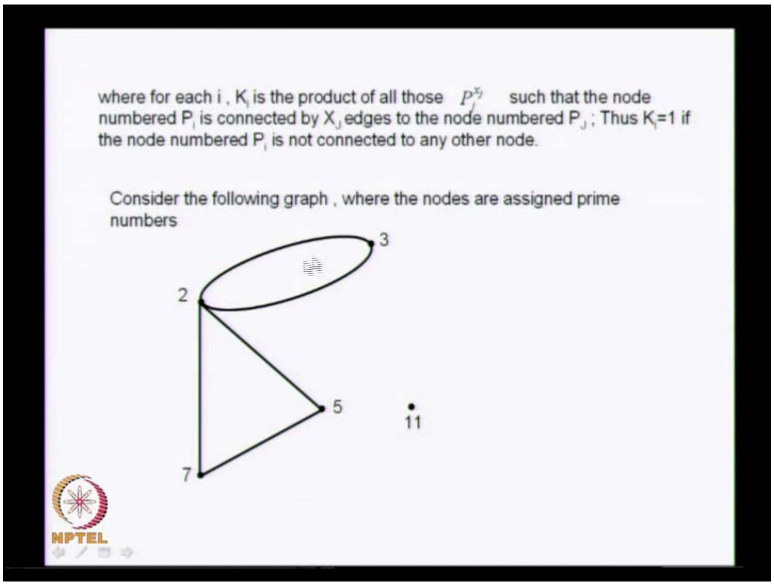
So, some sort of a cell preferential things, was there and he proved that you cannothave a complete in consistent system for integer arithmetic,any way we will not go into that now.

(Refer Slide Time: 55:09)



Next we will see what is the Godel number of a undirected graphs,any data structure also you can try to attach a Godel number to that. So, let us see how we can do it for undirected graph,an undirected graph consists of nodes and edges. So, we will not consider simple graph, we multipleedges can exist between 2 nodes. So, that the Godel number begin inwith the assignment of distinct prime number, to each node of the graphthe assignment is made arbitrarily.

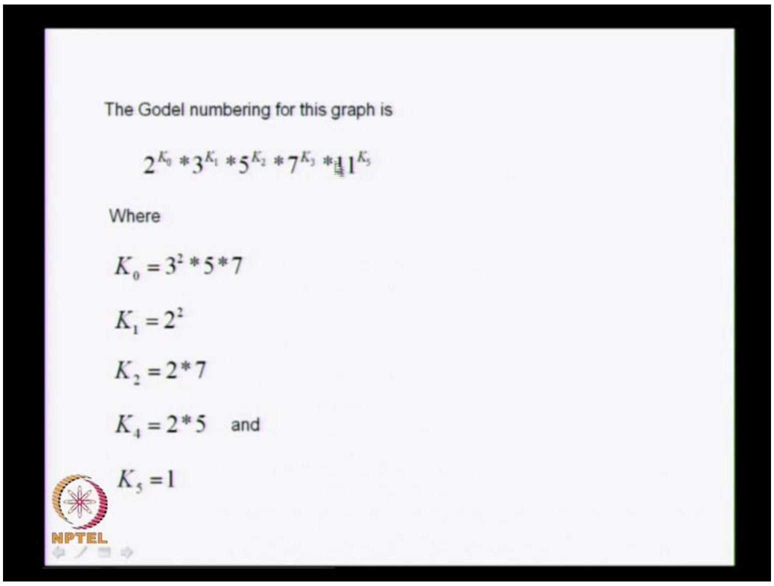
(Refer Slide Time: 55:44)



So, Ihave a graph with 4 nodesand Iam assigningprime numbers to this. So, 2 is assigned to this, three is assignedthis, 5 is assigned to this,7 and 11. So, you could have given in a different wayalso,this assignment is arbitrary and when we note thatwe have information about how many nodes there areand which nodes are connected to which by edges, we have a complete information about the graph.Supposing the nodes have been numbered P 0 P 1 and to P n, because we are assigning prime numbers to them.We take a Godel number of the graph to be the number P 0 to the power k 0, P 1 to the power k 1, P 2 to the power K2 and so on, up to P n to the power k n.

For example here we have 2, 3,5,7 and so onAnd we will have the corresponding Godel number, now when P j is raise to X j,it is actually K j,K j is theproduct of all this numbers.What is K i? K j is 1 if and only if the number P is not connected to any other node, consider the following graph where the nodes are assigned prime numbers2 is connected to 3 by 2 edges,2 is connected 5 by 1 edge,2 is connected to 7 by one edge.

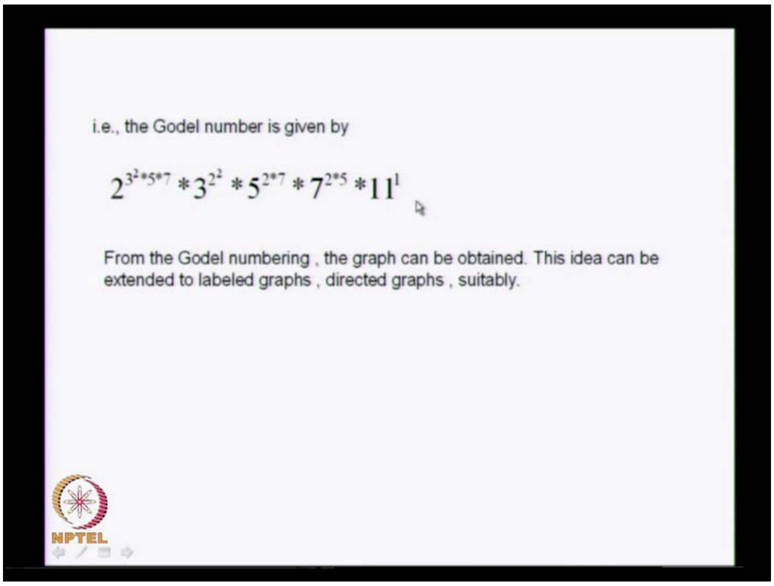
(Refer Slide Time: 57:23)



So, the Godel numbering for this graph will be 2 power k 0, three power K1,5 power K 2,7 power K3, 11 power k 5,where 11 is an isolated point,it is isolated node. So, K 5 will be 1, 2is connected to 3 by 2 edges, it is connected to 5 and 7 by one edge. So, K0 will be 3 power 2 star 5 star 7that ismultiplied by 5 and multiplied by 7,let us go back3 is connected to 2 by 2 edges. So, it will be 2power2.

Now, K 2is connectedfor the node 5,5 is connected to 2 and 5 is connected to 7. So, we have by one edge. So, 2 power 1 and 7 power 1, but 2 power 1 multiplied by 7 power 1 andthe fourth one 7 is connected to node number 2 and node number 5 by a single edge. So, you have 2 power 1 multiplied by 5 power 1 and that is the value of K 4. So, the Godel number of this graph is given by this numberand if you have assigned different numbers for this suppose I called this as 2 or called this as 5 then the Godel number will be different.From the Godel number of a graph we can get back the graph, but for one graph if we renumber the nodes we may get a different Godel numbering.

(Refer Slide Time: 59:16)



Now we can extend this concept toDirected graphs,Labeledgraphs and other data structures also.So, for the given graph in the figure the Godel number will be this much. So, thus we can have a Godel number for all data structures, defined in a suitable mannerwe are using this concept in the construction of counter machines.