Theory of Computation

Prof. Kamala Krithivasan

Department of Computer Science and Engineering

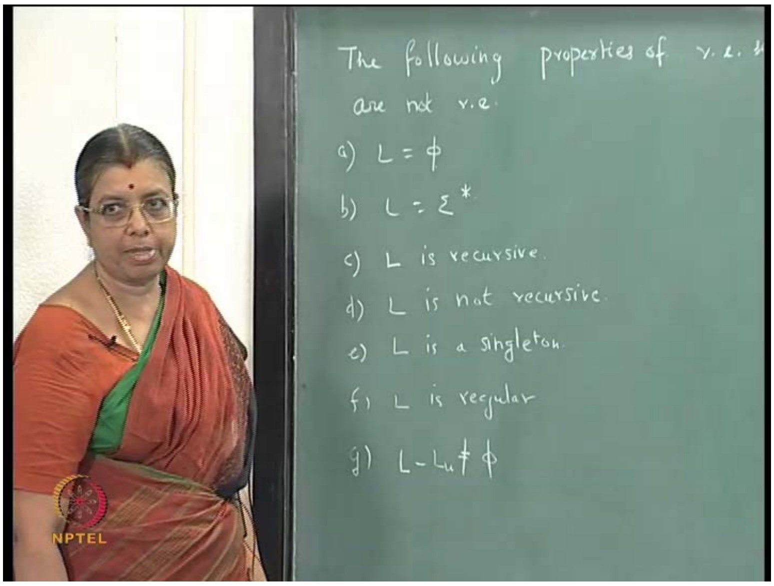
Indian Institute of Technology, Madras

Lecture No. # 34

Post's Correspondence Problem

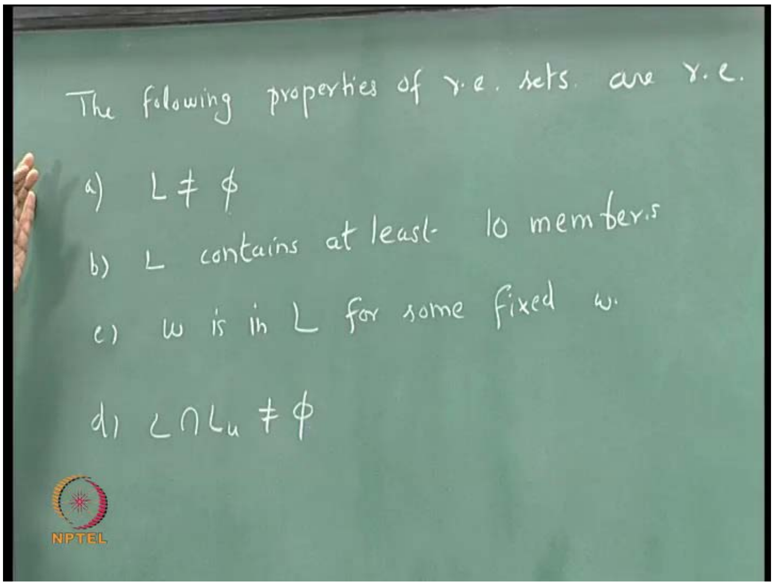
So, we were considering decidability results. If a problem has a corresponding language, then the problem is undesirable if the language is not recursive, but the language may be recursively enumerable or not recursively enumerable.

(Refer Slide Time: 00:36)



So, we have seen that some are recursively enumerable; some are not even recursively enumerable. Look at this problem is L empty is L equal to sigma star, these problems are such that the corresponding languages are not even recursively enumerable, they fall outside the recursively enumerable results. Is the language generated empty, is the language generated, is the language accepted by Turing machine equal to sigma star, is the language accepted by Turing machine recursive, is the language accepted by the Turing machine not recursive. Does it have just only one string accepted or is the language accepted regular or is the language accepted minus L u, you know what is L u m w m accepts w. Is it non empty, these questions are not decidable and also the corresponding language falls outside recursively enumerable results.

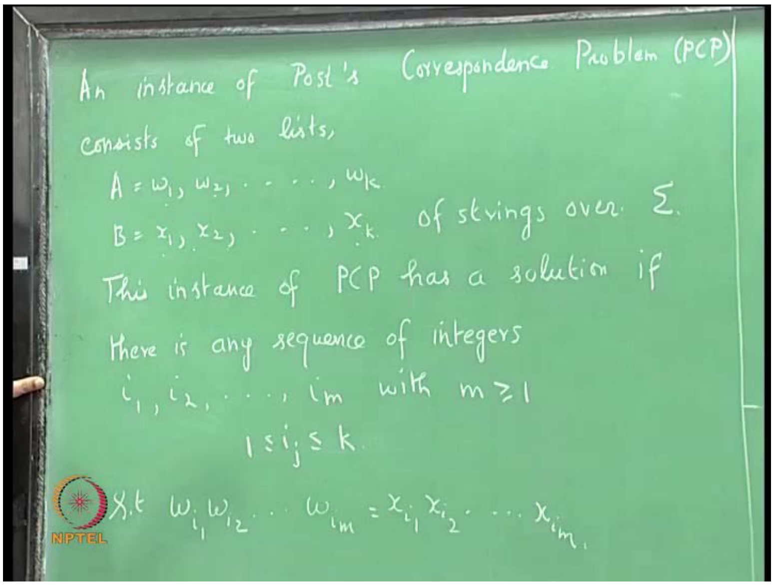
(Refer Slide Time: 01:42)



These properties of recursively enumerable sets are not decidable, but the corresponding languages, they are recursively enumerable they fall within recursively enumerable. They are not recursive, but recursively enumerable is L non empty or does L contain at least ten members is w in L. For some fixed w this is the membership problem, membership problem for Turing machines is recursively enumerable, but not recursive. For context sensitive languages it is peace face complete, for context free languages it is order encubed delgar some you have, finite sheet automata linear algorithm you have is L intersection L u not equal to phi. This is again recursively enumerable corresponding language is recursively enumerable, but not recursive

All this are undesirable problems, they are non trivial properties of recursively enumerable languages. So, by oasis theorem they are not decidable, but some are the languages if you consider that language what is that language? M the Turing machines for which the language L M will satisfy this, some are recursively enumerable, some are not even recursively enumerable. There is a theorem oasis theorem for recursively enumerable index sets, it is will not be considering. Now, that tells you under what conditions that language, corresponding language is recursively enumerable, but not recursive, there are three conditions to be satisfied and so on.

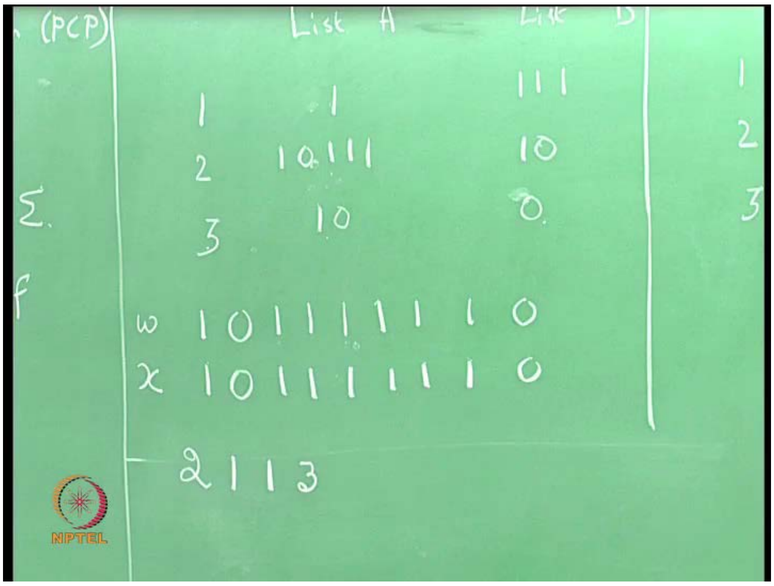
(Refer Slide Time: 03:44)



Next we go on to post correspondence problem, which is a another undesirable problem generally it is called p c p. So, what is a post's correspondence problem? This is again undesirable; we shall prove that it is undesirable. An instance of the post correspondence problem has two lists, it has got two sets of strings w 1, w 2, w k one set is that another, set is x 1, x 2, x k the same number of strings, the strings are different, but the equal number of strings over an alphabet sigma. We will assume that sigma has at least two elements, if it has got one element it is different it is a problem will become different, it has got only minimum two letters sigma has at least two letters.

And you have two sets of strings w 1 w 2 w k and x 1 x 2 x k, this particular instance of P C P you say that it has a solution. If you have a sequence of integers i 1 i 2 i m each is within 1 and k, such that w i 1 w i 2 w i m is equal to x i 1 x i 2 x i m you can catenate them in this order w i 1 w i 2 w i m. The string which is formed is equal to the string formed in the x i 1 x i 2 the strings are taken in the same order. If this is possible you call it, you say that this instance of P C P has a solution, if this is not possible the instance does not have a solution. So, the question is given two sets of strings does it have a solution or does it not have a solution. This is an undecirable problem, we will prove that.

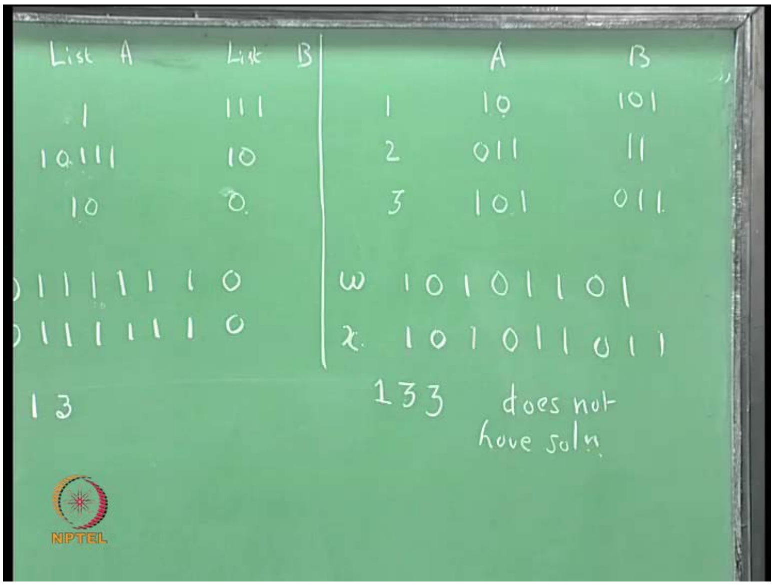
(Refer Slide Time: 06:00)



Now, as an example take these two lists, there are two lists list A and list B this has got three strings, this has got three strings. Does this have a solution, if it has a solution it cannot begin with this three because the first symbol itself is different there, it has to begin with this or this the probability looks like this is a better solution. So, if you start the solution taking two the w string is this, the x string is this. Now, after concatenating you must have the same string here and the same string here.

So, next you have a 1 1 1 here. So, it is it looks that better you take this here. So, you choose 1 now next pair chosen is 1. So, you add a 1 here, this 1 is added here, that 1 1 1 is added here, still they are not equal they match up to this point, but they are not equal you must have the same string in same w string and same x string. So, next again we choose 1, the same 1 here so, 1 you have chosen 1 1 1 here. Now, the x string is longer, then if you choose 1 0 0 here, 1 0 0 that is 3, the two strings become equal. So, this instance of P C P has a solution and what is the solution? 2 1 1 3 is the solution. 2 1 1 3 2 1 1 3 2 1 1 3 will be a solution, any number of times you take also will be a solution that does not matter. We take one, if there is one sequence which satisfies that is ok, we stop at that stage.

(Refer Slide Time: 08:36)

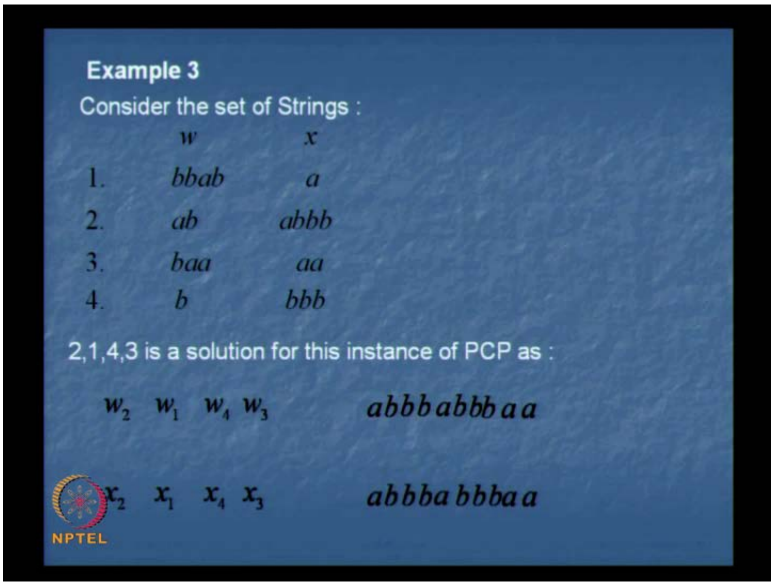


So, this instance of P C P has a solution and what about this instance of p c p? You have two lists A B each having three strings, this is a w list, this is a x list let us see whether this has a solution or not. Now, the w string and the x string you have to form, you see that you cannot start the solution with this because the first symbol is different, you cannot start the solution with this, the first symbol is different. So, if at all there is a solution it has to start with 1. So, start with 1 so in the w portion you have 1 0, in the x portion you have 1 0 1.

Now, in order to continue the solution I must have the w string beginning with the 1, this symbol has to be 1 here to continue with the solution. So, I can have this or this I cannot have this. So, if I cannot have this suppose, I take 1 0, then the B 1 is 1 0 1, there will be miss match here. So, I cannot continue the solution with (No Audio Time 09:59 to 10:07) 1. I have to use 3 only, if I use 3 I will get 1 0 1 0 1 1, the two strings match up to this point, but the x string is slightly longer.

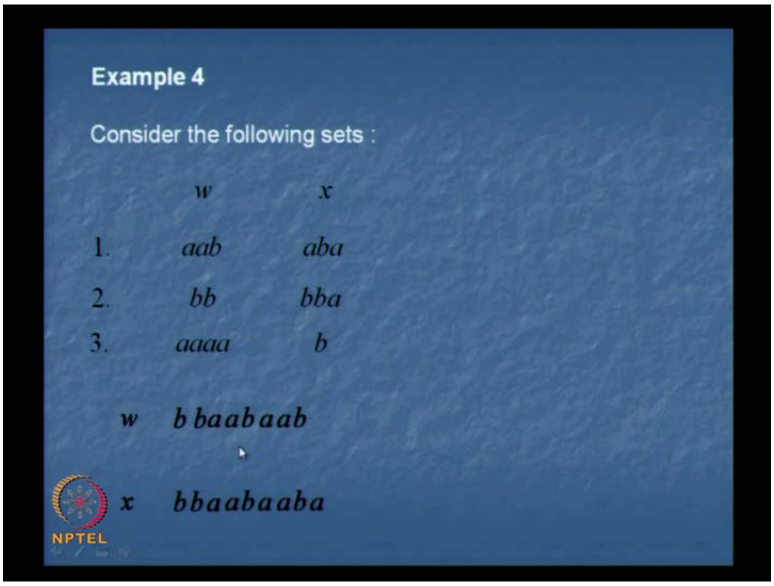
Now, the same situation has come, if I have to continue the solution. I have to again take 3 only that is 1 0 1 0 1 1. Any number of times I can choose 3, the two strings will match after a point, but x string will be longer one more 1 will be there, that will be always there. So, this that instance does not have a solution, this has a solution which is 2 1 1 3.

(Refer Slide Time: 11:17)



We have considered two examples, two instances of the post correspondence problem, we shall consider two more examples. Consider the following sets of strings you have four strings and you have four strings in the x set. Now, does this instance of P C P have a solution, we find that 2 1 4 3 is a solution for this instance of P C P as you see starting with w 2 and x 2 you have this string. Then proceeding with w 1 and x 1 you get this, after this point they are equal, but first string is one symbol more, continuing with w 4 and x 4, the solution is this after this point. Now, the lower string is one symbol more, now you have w 3 and x 3. Now, you find that you have the same string in w and in x. So, 2 1 4 3 is a solution for this instance of P C P.

(Refer Slide Time: 12:30)



Let us consider one more example, consider the following sets with three strings in w and three strings in x. If a solution exists here, it cannot begin with 3 because the first symbol is a here and the first symbol is b here, it cannot begin with 1 also because if you start with this, the first symbol is the same but the second symbols are different. So, if at all there is a solution, it should start with 2, if you start with 2 we get this. Now, to continue the solution the w string has to begin with a because the third symbol is a here. So, either it can continue with this or it can continue with this, but it cannot continue with this because if you take a a a a here, then the fourth symbol will be a a and the fourth symbol will become a b here.

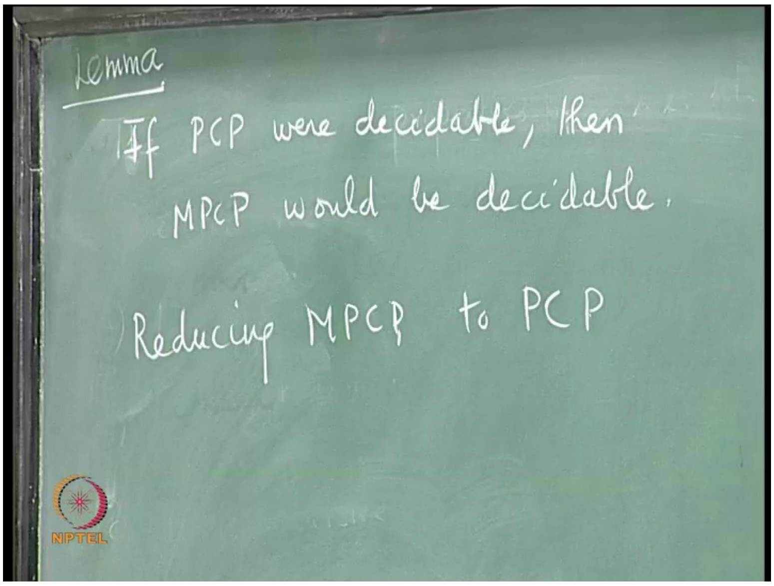
So, the solution cannot be continued with this. So, it has to follow this one so, continuing the solution with 1 you get this. Now, the two strings match up to this point, but the string x is longer by one symbol. Again the same situation occurs you cannot continue the situation with three, it has to continue with one only. Now, if it continues again the same situation occurs and at known time, the string w will become equal to the string x. So, for this instance of post correspondence problem no solution exists.

(Refer Slide Time: 14:28)

So, the question is given an instance the this is the problem. Given an instance of P C P does it have a solution? This is not decidable and how do you prove that this is not decidable? (No Audio Time 15:03 to 15:16) Before going into that, we define a modified version of P C P. An instance of modified post correspondence problem is this, it has got again two lists A with k strings, B with k strings over an alphabet with at least two elements. This instance of M P C P you call it as an, M P C P has a solution, if there is a sequence of integers 1 i 1 i 2 i m. Such that w 1 w i 1 w i 2 w i m is equal to x 1 x i 1 x i 2 x i m.

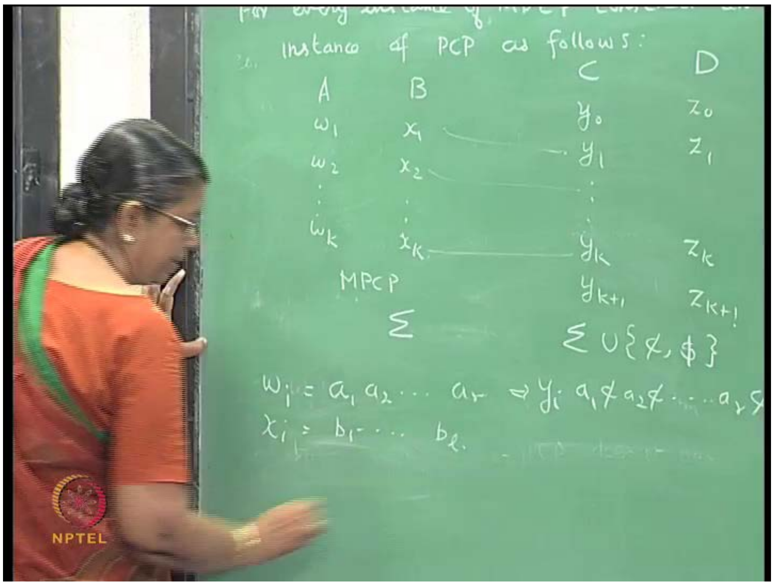
So, you modify the P C P a little bit, modified P C P has again two lists with k strings each and this modified P C P has a solution. If you can find a sequence of integers 1 i 1 i 2 i m, such that w 1 w i 1 w i 2 w i m is equal to x 1 x i 1 x i m. The difference between P C P and modified P C P is that the first string is fixed. The solution where you start, you have to start with w 1 and x 1 that is fixed, then you can continue with the solution, that is not there in P C P. Actually it is not very much different because you can convert one instance of M P C P into P C P that is what we will see next.

(Refer Slide Time: 17:10)



What we prove is. If P C P were decidable, then M P C P would be decidable, we will prove this you prove this is a lemma, if you can solve P C P you can solve M P C P, this is a lemma or in other words you are reducing M P C P to P C P this is how you say, you are reducing M P C P to p c p. Now, in order to the difference between them is that the first string is fixed in M P C P, you have to start the solution with w 1 and x 1, then you have to find integers i 1 i 2 i m such that the you get equal strings.

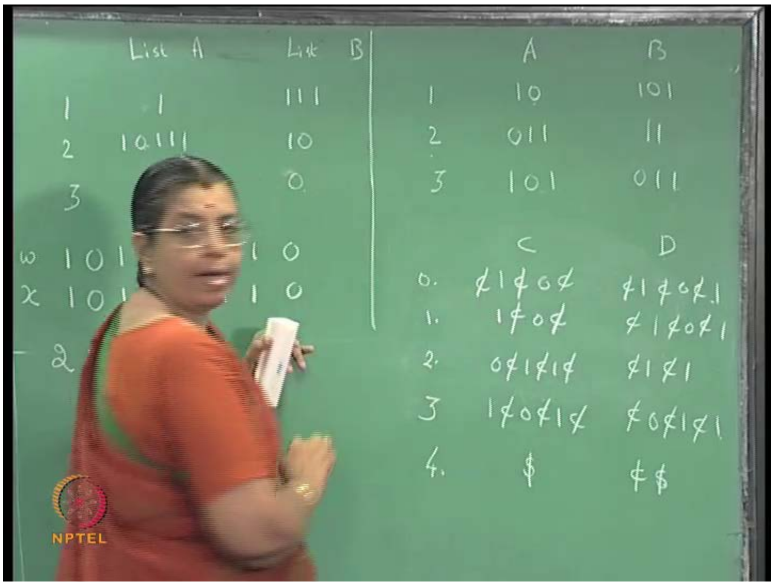
(Refer Slide Time: 18:50)



Now, for every instance of M P C P you construct an instance of P C P as follows. (No Audio Time 18:36 to 18:50) Now, you have two lists A B w 1 w 2 w k x 1 x 2 x k. Now, from this, this is the instance of M P C P now, from this you construct two list C and D. The strings are y naught z naught y 1 z 1 y k z k y k plus 1 z k plus 1, if each list has k strings here each list will have k plus two strings. And from this how do you get this? From this how do you get this? From this how do you get this? Then we will see what is y naught, what is y k plus 1.

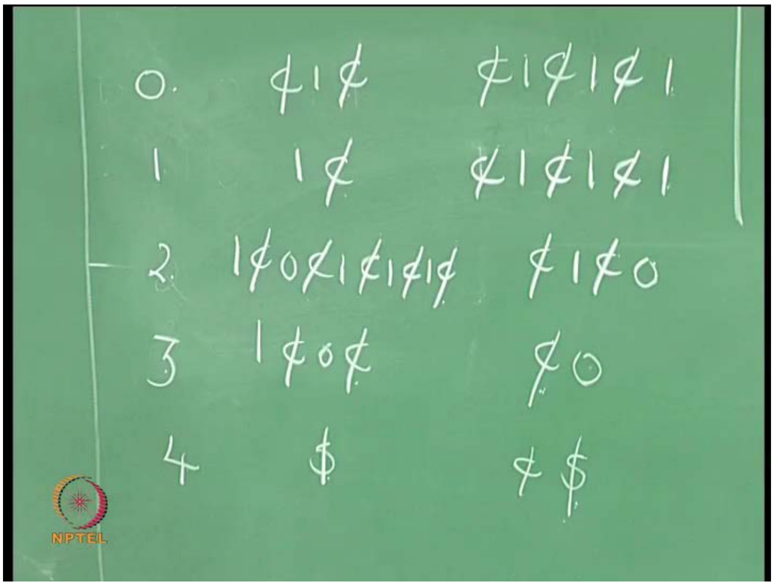
Now, the alphabet is sigma means here the alphabet is sigma union two more marker symbols, you have two more symbols, which are used as markers. Now, how do you get y 1 from w 1? (No Audio Time 20:52 to 20:58) Suppose, w 1 w i I will say w i is equal to some a 1 a 2 a r, then this implies y i will be a 1 marker sign, a 2 marker, a r marker, after every symbol use the marker. And if x i is b 1 b 2 b l say, then z i will be marker b 1, marker b 2, marker b l, every symbol is preceded by the cent sign marker sign.

(Refer Slide Time: 22:26)



So, for example, if I take this instance, if you want to form C D lists, there are three strings in each list here there will be three plus two five strings. So, the first one we will write later, second one will be every symbol is followed by a cent sign here, every symbol is preceded by the sign here. So, it will be the first one zeroeth one we will see how it is, cent 1 cent 0 cent 1. So, again one will be 0 1 1 so. 0 1 1 and here it will be 1 1. So, 1 1, third one will be 1 0 1, 0 1 1, the last one is dollar here and cent dollar here. The first one this is the same as this c 1 c 0 c 1, here this is preceded by a cent sign c 1 c 0 c, this is the same as this with 1 c added.

(Refer Slide Time: 24:05)



Let us write for here, the zeroeth one will be (No Audio Time 24:06 to 24:13) one is this 1 followed by the cent sign, here it will be preceded by the cent sign, here this each symbol will be followed by a cent sign, here each will be preceded by a cent sign, here each one will be followed by a cent sign. This will be preceded by a cent sign the last one is dollar here and cent dollar here.(No Audio Time 25:04 to 25:10)

Now, you see that, if the solution has to start the first symbols are all different here, if the solution has to start it has to start with 0, look at it as a P C P, this is an instance of M P C P. So, the if at all the solution is there it has to start with 1, if similarly, here also. So, if the solution has to start it has to start with this, it has to end it has to end with this, the last symbol here is cent here it is some other symbol say is a solution has to end it has to end with the 4 this one .

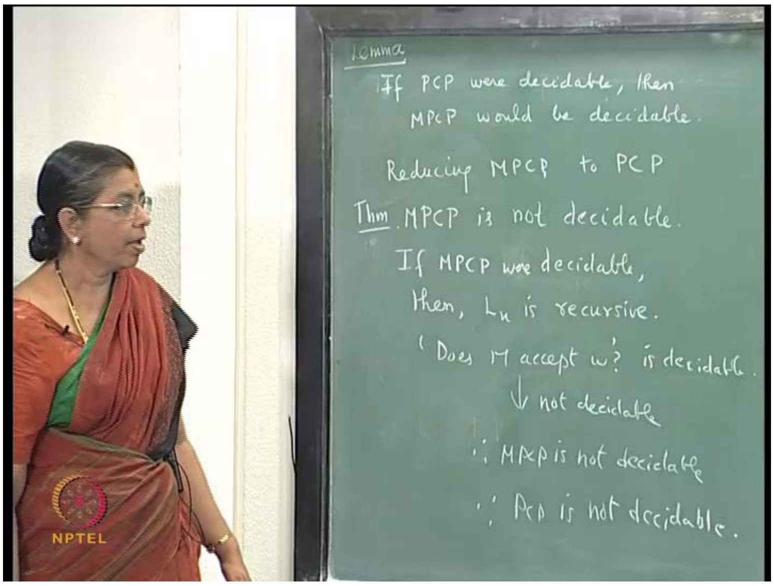
So, in general what happens is from two lists A B you have formed two lists C D, corresponding to w 1 x 1 you got y 1 z 1 by writing cent sign after each symbol here and before each symbol there. And the y k plus 1 z k plus 1 is dollar cent dollar y naught and z naught are such that z naught and z 1 are the same, but y naught is y 1 with one more cent symbol in the beginning. If the solution has to, if you look at this as the P C P, this is M P C P, this is this the P C P we have constructed, if you look at it as a P C P, then the solution has to start with 0 and end with k plus 1. So, it has to start with 0 and end with k plus 1.

(Refer Slide Time: 27:44)

Now, if you find sequences, such that 0 i 1 i 2 i 3 i m k plus 1 is a solution here. Can you have a solution for this M P C P? The first string alone you are considering with one additional C here is not it. So, if it starts means the corresponding string is 1 really, then for i 1 again you choose the same i 1 i 2 i m, the last is to adjust the markers and have the dollar sign is not it. So, if this is a solution for the M P C P, this will be a solution for the P C P. And if you, this is the solution for the P C P, this will be a solution for the M P C P.

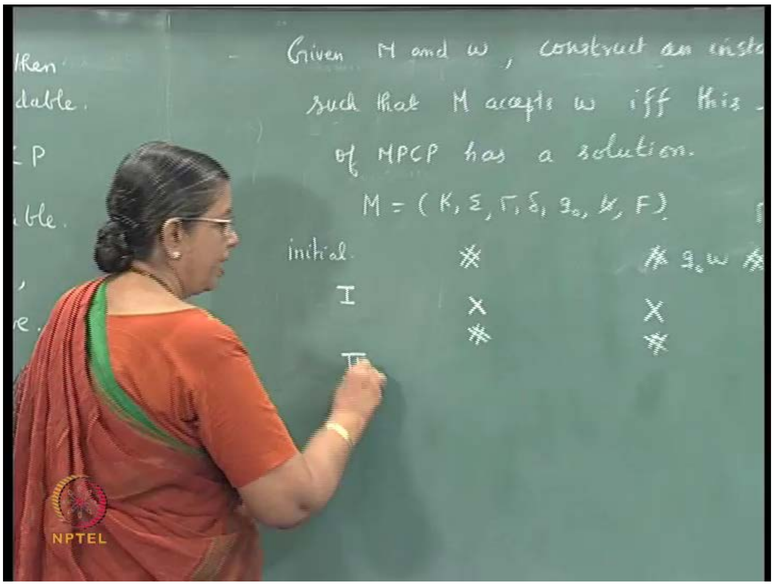
So, any instance of M P C P, you are able to; for any instance you are able to construct an instance of P C P. Such that the M P C P instance has a solution if and only if the P C P instance has a solution. So, given an M P C P instance you can construct a P C P instance and if P C P were decidable, you will be able to say yes or no then for instance of M P C P also, you will be able to say yes or no. That is why you say, if P C P were decidable, then M P C P would be decidable.

(Refer Slide Time: 29:15)



The next thing we have to prove is P C P, M P C P is not decidable (No Audio Time: 29:28 to 29:39) or you say that. If M P C P is decidable, then L u is recursive or the question. Does M accept w? Is decidable you will prove this. If M P C P were decidable (No Audio Time: 30:18 to 30:29) then the question does M accept w is decidable, but you know that is not decidable. Therefore, M P C P is not decidable. Therefore, P C P is not decidable. So, you are reducing L u or this question does M accept w to M P C P and you are already reduced M P C P to P C P. Now, how does the proof.

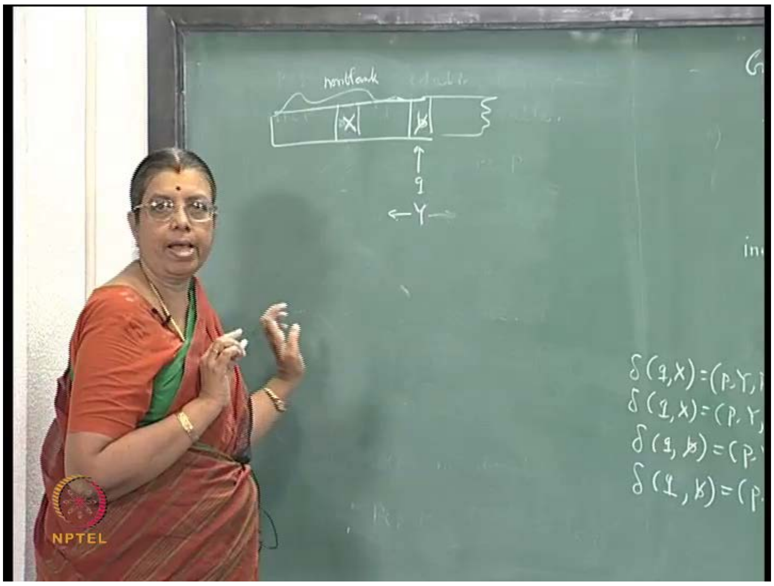
(Refer Slide Time: 31:20)



Given M and w given a Turing machine M and string w, you construct an instance of M P C P. (No Audio Time: 31:43 to 31:50) such that, M accepts w if and only if, this instance of M P C P has a solution. The proof goes like this, given M and w, we construct an instance of M P C P. Such that, M accepts w if and only if, that instance of M P C P has a solution. So, M is given by k sigma, gamma, delta, do not, blank, F. Now, the Alphabet where there M P C P Alphabet will be gamma union k union hash symbol one more symbol. All type symbols states and an hash symbol, that will be the Alphabet for the M P C P. Now, the paths will not write 1 2 3 but will write depending on the mapping of M.

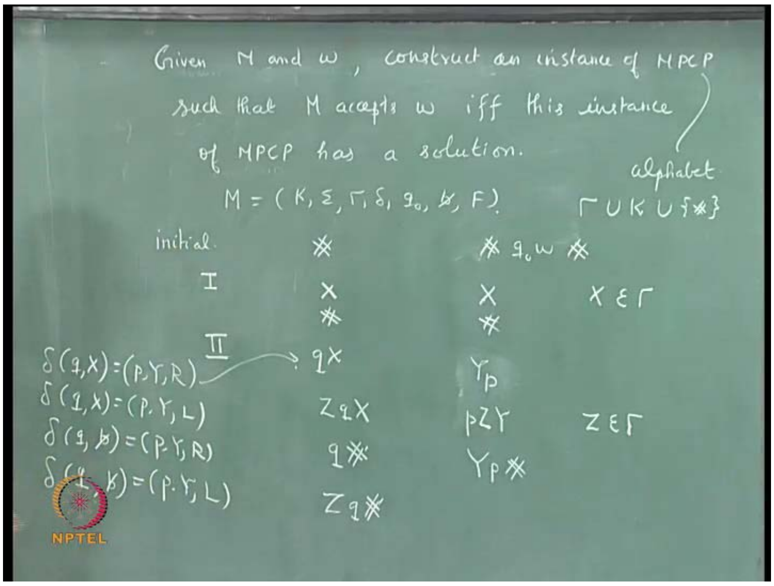
So, we will divide them into group initial M P C P means the first one has to be mentioned, initial one first one which how you have to start the solution? The solution will be hash q naught w hash, just hash q naught w is the initial ID. If you remember for w when you have to start what is ID initial ID? q naught w. So, that the initial ID is put between hash that is two hashes and first this pair is just hash, then you have some paths. Group one will be (No Audio Time: 34:32 to 34:37) you have x, you have x here, where x belongs to gamma and then you have, can have hash or hash also u will have but not the states.

(Refer Slide Time: 35:35)



Group two is to simulate, the Turing machine. In Turing machine you can have left moves, you can have right moves and we have to distinguish two cases in the right moves. One is when you read a symbol, one when you read a first blank. (No Audio Time: 35:22 to 35:34) So, the tape may be like this, this is the non blank, this is the first blank. This is the non blank portion you may be reading a symbol in state a q or I will use the symbol x or you may be reading this first blank symbol you have to distinguish between these two cases.

(Refer Slide Time: 36:26)

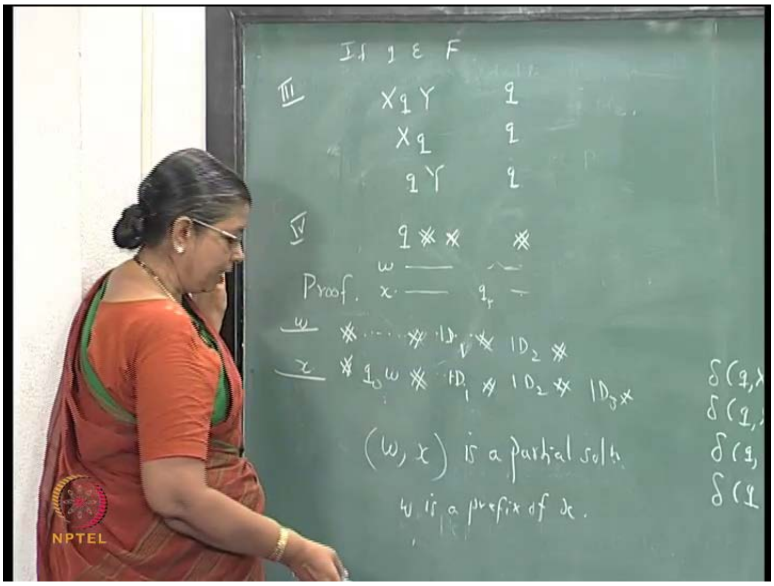


Now, if delta of q x is equal to P Y R this could be the move, then you have the pair q x, y p for this you have the pair q x y p. And you can also have the move delta of q x is equal to P Y L left move, for this the corresponding pair will be z q x and p z y, there what is z? Z can be any symbol, z belongs to gamma any symbol, it can be any symbol. So, for each symbol you will have a pair, for each symbol is there you will have a pair, here right move for one mapping you have only one pair, for left mapping for each symbol of gamma you have a pair. There will be several pairs like that, then the blank symbol, if it is reading this position.

If it is reading this first blank symbol, then delta of q blank is equal to P Y R it could to be the mapping can be like this, it can print y here and then move right that is possible. In that case the pair will be q hash is y p hash, it is printing y and moving right. You can also have delta of q blank equal to P Y L, it can be a move, what it does it prints a y and moves left in state p, here again for each symbol of gamma you will have a pair. So, it will be z q hash, then it prints y, but moves left and it will read a z in p this is for each z.

So, it is a left move there will be several pairs or pairs equal to the number of symbols in gamma, right move there is only one pair. So, these are the pairs, which you write for the moves of the Turing machine, given w and m you have to write the two lists A and B. And the first to it because it is M P C P the first pair you have to specify, this is the first pair, then for any symbol in gamma you have x x hash hash and then simulating the moves you have this. Then once it accepts, it will accept in a final state.

(Refer Slide Time: 40:57)



That is the third pair, if q belongs to f once you reach the final state the pairs you have will be x q y q you start consuming the symbols actually. So, x q is q q y q, this is for consuming the symbols. The last pair will be q hash hash hash the proof is like this, you start the solution with initially like this start q naught w is not it, the solution will start like this. Now, you have to pick pairs such that you get q naught w and hash here, you have to pick the pairs, but the pairs have been written in such a way that they are simulating most of the Turing machine. If q of x x is reading x is being read instead q, the next instance y is printed and the state changes to p and it is moving.

So, in the I D, if you have q x something the next instance what will happen? This portion will remain as it is and this portion also will remain as it is, x changes to y and q changes to p this will be the next I D. So, this is changed into this and similarly for left move suppose, you have this and you have a symbol z here, the next instance it will change to p z will be read this will be changed to y is not it. So, this is the pair you are getting.

So, while you try to fill this portion with q naught w this, the next I D will come here and now, you have to fill this with this string this pair. Now, when you try to do this, the next I D 1 I D naught this is I D naught this is I D 1 I D 2 will come here, when you try to fill this with I D 1, then next when you try to fill this with I D 2 you will get I D 3. This is w x the string w this is the string w, this is the string x, this is such that, this is the partial solution you say it is a partial solution, if w is the prefix of x.

Now, you continue this solution, this w x you continue the solution until you reach a final state. Then when you reach a final state, symbol by symbol you try to absorb. So, ultimately here you have a longer string you have a shorter string here. So, step by step you do it and ultimately you will end up with equal strings. And these are defined the consuming symbols is defined only when you reach a final state. So, only when you reach a final state, you will start consuming the symbols and you will end up with equal strings. (No Audio Time: 45:10 to 45:22)

(Refer Slide Time: 45:31)

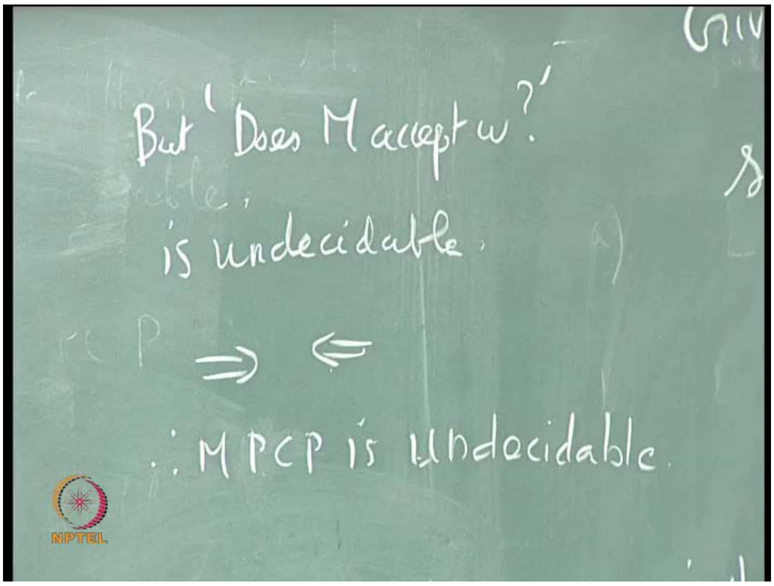
So, in the proof you can use induction and say that w x is a at any stage w x is a partial solution. (No Audio Time: 45:38 to 45:50) That is the solution is of the form I D 0 I D 1 I D k minus 1 and here, you will have the same thing plus I D k. And this, for this portion you can use induction, because initially you start with hash and hash q naught w hash and then you proceed. Then once you reach the final state you start consuming the symbols and it will be possible to end up.

So, what is the solution you have got the, from given w and M, you have constructed an instance of M P C P. And if w M accepts w then this will have a solution, you go to a final state and then you will get a equal string. If M does not accept w you will not go to a final state you may halt in that case. Again that consuming of symbols will not be possible you will not get equal strings, if it gets into a loop if m gets into a loop and w. This partial solution will keep on continuing and ultimately you will not reach a final state it will go on and on and on, you will not be able to get equal strings.

(Refer Slide Time: 47:56)

So, M accepts w if and only if this instance of M P C has a solution. Suppose, M P C P is decidable, then what you can do is given M and w. You construction construct a name instance of M P C P and give it to the algorithm it is also yes, this will come out with a answer yes or no. If it is yes, then M accepts w, if it is no, M does not accept w. So, you are able to solve the problem does M accept w, but you know that, that is already an undecidable.

(Refer Slide Time: 48:58)



So, assuming that M P C P is decidable you are arriving at a contradiction. So, but this problem does M accept w is undecidable, you know that it is undecidable. So, we are arriving at a contradiction. Therefore, M P C is undecidable and earlier you have proved that, if P C P were decidable M P C P was decidable, but M P C P is not decidable so, P C P is not decidable therefore, P C P is. So, we have reduced the problem does M accept w to M P C P and we have reduced M P C P to P C P ensuring that all these problems are undecidable. So, we will take an example in the next class and also show how the problem whether the grammar context is grammar is ambiguous or not is undecidable making use of this.