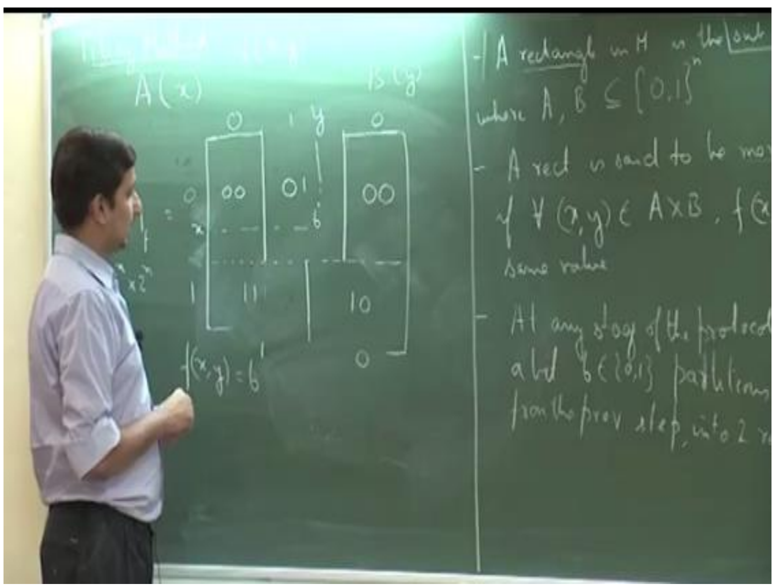
Computational Complexity Theory
Prof. Raghunath Tewari
Department of Computer Science and Engineering
Indian Institute of Technology, Kanpur

Lecture -41
Introduction

So, today the only new thing that we will see is another way of lower bounding the communication complexity, and in fact we will also see that this method is more powerful than the fooling set method that we saw on Monday.

(Refer Slide Time: 00:38)



So, this is known as the tiling method, this is also a very intuitive approach with a nice argument, so suppose you have a communication complexity protocol between Alice and Bob where let us say that Alice has x and Bob has y, so what we do is to bound the communication complexity of some function that they are computing. So, let us say they are trying to compute some abstract function f x, y.

So, we build this matrix, so this is a, so we call this matrix M of f, but when f is clear will just denote it by M itself. So, M f is a 2 to the power n cross 2 to the power n matrix where the rows of the in the matrix this matrix M is indexed by the x's and the y the columns are indexed by y's.

So, since we are looking at n bit strings there are 2 to the power and rows and columns each and we set the x, y entry as let us say the bit b if f of x, y is equal to b.

So, it is a 0 1 matrix, so now the question is how do we study the compute the communication complexity by studying this matrix? So, let us look at some basic definitions, so a rectangle in M is the sub matrix A cross B, where A and B are some subsets of 0, 1 to the power n. So, basically you pick some subset of your rows and some subset of your columns and then define the cross product so that basically defines a rectangle.

So, in general, if you look at the set of all x's and all y's that will give you the matrix M but also any subset will give you any proper rectangle. A rectangle is said to be monochromatic if for all x, y belonging to that rectangle f x, y has the same value. So, it is either all 0's or all 1's. So, now why do we need this thing, so let us see what happens, so suppose initially Alice communicates a bit to Bob.

So, let us say Alice communicates b1, so what happens to the rectangle as a result of that communication so one way to view this is the rectangle basically gets divided into two halves. So, let us say that we have one half of the rectangle which corresponds to those x's for which Alice will communicate a 0, so let us say this communicates this corresponds to Alice communicating 0.

And maybe we have another half which corresponds to those x's for which Alice communicates a 1 and it need not be I mean I drew it like this, but it need not be the case that all these x's for which Alice communicates a 0 are contiguous and so are the 1s I mean it can be that for this x Alice communicates 0 for the next one maybe for the next 0 and so on, but that does not matter by the definition that we have.

But the more important point is that for every bit that Alice or Bob communicates the rectangle or each rectangle from the previous step gets partitioned into two new rectangle. So, at any stage of the protocol communicating a bit let us say b in 0, 1 partitions each rectangle from the previous step or stage as I said into two rectangles. So, in other words so suppose that this is what was the first bit? I mean; suppose this was the partition.

That resulted by communicating the first bit, so now Bob decides to communicate his bit, so let us say for these y's Bob communicates say zero and also maybe for these y's and for this middle portion Bob communicates a one, So, this rectangle gets so this top rectangle now gets partitioned into these two rectangles. So, think of this as one rectangle and this has the other rectangle, and the same thing can happen for this also.

Maybe here, I mean if Bob had a 1 in the first stage then maybe this is what he does for I mean this is the bit that he communicates for these strings and this is the bit that he communicates for these strings, so basically and this thing will continue then again in the third stage each of these rectangles will get divided into two more. So, the point is that so what do each of these rectangles correspond to.

So, this rectangle corresponds to those pairs of x and y where the communication pattern was 0, 0 this corresponds to those strings those pairs of strings for which the communication pattern was 0, 1. What about this? Again 0, 0 this was 1, 1 and this was 1, 0.

(Refer Slide Time: 09:21)

So, what we can say is if a protocol communicates k bits then we have 2 to the power k such rectangles. And the other property that these rectangles have is that so suppose if I am communicating suppose if I look at a protocol which communicates in the following manner what can we say about the entries in each such rectangle? Can I have a different entries? So, at the end of the protocol.

So, let us say that, so when the protocol ends what can I say about the entries of each matrix they will all be the same. So, either for a particular rectangle they are all 0s or all 1s because I mean basically what that correspond what each rectangle corresponds to, is a particular sequence of bits that gets communicated or a particular communication pattern. So, therefore the final output that occurs is also the same.

So, when the protocol ends each rectangle has or each rectangle is monochromatic. So, let us define let us look at the following definition, so we are assuming that a protocol correctly computes the function, M of f is basically so we are looking at this matrix for the sake of proving the lower bound. So, m of f I mean Alice and Bob do not have access to M of f, because let us see if one of them had access to M of f he would just look at that particular entry and output it.

I mean they have access in some sense but they do not know what the other guys input is Alice does not know what y is and Bob does not know what x is, so this is just an abstract matrix that we can think of such that given a function every entry of this matrix has some binary value, monochromatic property just means that all the entries are the same in that set, so you might divide.

I mean basically see division happens as a result of one step of the protocol, so maybe Alice and Bob all I mean even after the rectangles become monochromatic they may proceed further but that is not the point, the point is that when the end. So, when Alice and Bob end in other words when they correctly output the value of f of x, y at that point the rectangles that we get as a result of this definition are all monochromatic.

And the reason for that is because, suppose that one of these rectangles is not monochromatic hat does that mean? That means that there are two values I mean there are two entries in that rectangle that have different values, but if that is the case what that means is that, for the same sequence of communication pattern between Alice and Bob on one input they are outputting 0 and on another input they are outputting a 1.

But that cannot happen, and if they have the same sequence then they must have the same output. So, they do not care whether it is monochromatic or not, so think of this as an algorithm so Alice and Bob are two algorithms. So, these algorithms do the following so initially the algorithm of Alice takes x and it produces a bit, Bob takes y and the bit that Alice gave and produces another bit b 2.

Then in the third step Alice will take x b 1 and b 2 and produce a bit b 3 and they will keep on continuing in this fashion up to some t steps and the algorithm is a correct algorithm in the sense that we say that the communication protocol correctly computes a function f of x y if the last bit that is communicated either by Alice or by Bob I mean we do not restrict who communicates the last bit is equal to the value of f of x, y.

So, that is just an algorithm, so suppose if I ask you to design an algorithm that checks if two graphs are isomorphic. So you design an algorithm which should have the property that finally when the answer, and when the algorithm says yes that should correspond to two graphs that are isomorphic and if the answer the algorithm outputs no, that should correspond to two graphs that are non isomorphic.

So, the protocol is basically the algorithm, but in this case we have two algorithms that interact between them and the output is some fixed deterministic value. The matrix is just to help us formulate the proof as to how the communication complexity can be lower bounded, so that will just come in a moment. What do you mean by step here? B 1, some B 2 no it always gives 1. So, in n steps it gives some n bit.

No, the computations are dependent, so basically if I fix an x, y yes maybe for different x, y pairs maybe they are different. But suppose if I fix an x, y. So, the computation of Alice and Bob are deterministic. So, way to think of this is that so maybe so, so let me use the notation A to denote the algorithm of x. So, A given x produces a bit b 1. So, this b 1 is fixed if you fix x suppose x and y are fixed then there is exactly one b 1 that is getting outputted.

So, what does Bob do? Then Bob basically takes y and this bit b 1 and produces a bit b 2, so can you have any ambiguity in what b 2 is going to be, in the sense that can there be two values for b 2 for the same y can that happen, because if y is fixed from the first step we know that b 1 is fixed, so if y and b 1 both are fixed then b 2 has to be fixed. So, now again when you come to the third step Alice takes x his own bit b 1 may be it does not need that but let us just keep it.

And then b 2 and then produces some b 3, so again by our assumption x is fixed, b 1 is fixed, b 2 is fixed, so b 3 have to be fixed. So, that is why in t steps if the bit that gets outputted is b t that is some fixed bit. No it will always be the same the algorithm is the same, so that may be in that sense maybe yes so maybe the first firstly the algorithm checks how many parameters are provided and then based on that it will do its computation.

So, you can just think of, so if you are comfortable with that so maybe these are different algorithms A 1, A 2, and so on but the final algorithm A of Alice is just a union of all these algorithms. So, it will check its input and then depending on how many parameters it has it will apply that particular a I, so it is the same thing whether you assume that there are different algorithms or just one algorithm.

So, what was I going to say, so a monochromatic tiling of m is a partition of m into monochromatic rectangles and we denote chi of m or f, so we say that chi of f is equal to the minimum number of rectangles in any monochromatic tiling of M of f. So, we say that this matrix I mean there is a monochromatic so we say that some partition of this rectangle is a monochromatic tiling.

If all the entries in each rectangle have the same value and we denote chi of f as the minimum number of rectangles in any monochromatic tiling of f, so basically we look at all possible monochromatic tilings, and then for whichever monochromatic tiling the number of rectangles is minimized that is what the number chi of f is, chi f entirely depends on f it will not depend on any protocol because this is the minimum overall protocols in some sense.

So, this is one rectangle that is counted as 1 because see, because I can define I mean you can count them as 2 also as you like, but I will say that I mean why should i count them as 2, I mean i can just define my subset a to be these elements and I will define my subset b to be this union this. So, if I am getting a tiling, which will have one rectangle for those values I mean why will I count it twice because anyway i want to minimize it.

2 to the power n cross 2 to 3, in each step they are communicating 1 bit, so Bob knows that so let us look at it this way so when Alice communicates a bit b1 to Bob initially before that Alice could have any of these strings as its x, but then depending on what b 1 is Bob now knows that if that bit was 0, now Bob knows that only these strings are possible candidates for x not necessary but here I have that case;

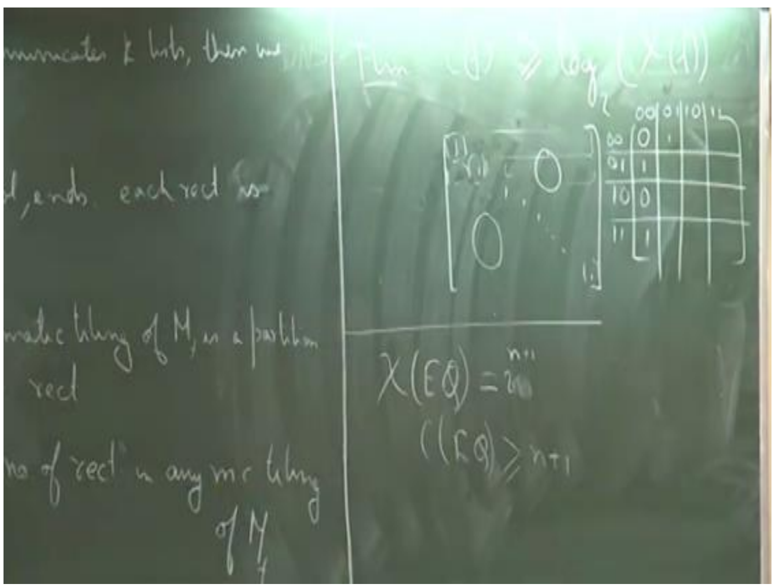
But depending on what b 1 is Bob, will know that what is the set from which Alice has his x and if b is 0 it is just the complement of that every entry has an address yes, no so these two guys are all powerful I mean see we do not have any computational restriction on Alice and Bob, in other words they do know the algorithms of each other also. I mean since so since they know what the algorithms of each other are they know this entire M f also.

But they do not know what is the particular argument on which that; algorithm is getting applied so Alice knows what the algorithm of Bob is and Bob knows what the algorithm of Alice is, in other words they know everything, the only thing that they do not know is each other’s input. So, suppose we are trying to decide, if let us say two graphs are isomorphic and let us say we have some communication complexity protocol.

So, I know what your algorithm is, and you know what my algorithm is but we are given the two graphs separately. So, you have one of the graphs and I have one of the graphs and we do not know what these graphs are so now we are trying to communicate bits between ourselves in order to decide whether the two graphs are isomorphic or not. So, as you said that the matrix is known to both Alice and Bob.

It is only how it is getting partitioned that is not clear to begin with graph isomorphism. I do not know of any non-trivial communication complexity protocol. I mean of course you have some protocol you will always have a protocol using n bits and one of the guys gives his entire input to the other guy, so they are all powerful they can do anything but whether you can do it in better than that I am not aware.

(Refer Slide Time: 27:21)



So, now once we have this what we can show very easily is that the communication complexity of this function is lower bounded by what do you think, it will be will be log to the base 2 of chi of f and what is the reason for that so the reason is again basically comes from the fact that it gets divided in into two rectangles at each step, I mean each rectangle. So, suppose 2 to the power n less than or equal to n it is always less than n, chi f of course chi f is less than 2 to the power;

That is 2 power n because this is lower I mean upper bounded by n plus 1 actually 2 to the power n plus 1 to be more precise because this is upper bounded by n plus 1. So, n plus 1 is an upper bound on this quantity, so 2 to the power n plus 1 is an upper bound on this quantity, no so if you recall last class what we said was that so suppose if Alice communicates his entire x to Bob then Bob has to give the final answer so that is why that extra bit is required.

No it is greater than or equal to log chi f, no so I mean let us, so both so let us see this, so intuitively what this means is that suppose I have a function, so now given the function I have fixed values on all the entries. So, now suppose if I consider any protocol between Alice and Bob that exactly computes that function so what should happen is that so if the protocol takes let us say k steps at the end of those k steps every rectangle should have either all 0s all or 1s.

So, in other words every rectangle must be monochromatic, so if every rectangle is monochromatic at the end and the number of steps taken is k the number of total rectangles is 2 to the power k. So, suppose if I already have a some way of assigning values, to this matrix such that the minimum number of monochromatic rectangles is let us say some chi of f then the communication I mean any communication protocol will take at least log chi of f many bits;

To correctly output f so that is the argument, I did not understand so what are you saying say that again see you can always make a very trivial monochromatic rectangle where you say that well each rectangle is just one entry, so then you always have how many 2 to the power 2 n many rectangles. For each column you can have more than two rectangles also, so you are so for each column you will have two rectangles you can always define it that way.

So, that will give you upper bound of 2 to the power n, then 2 to the power n plus 1 because 2 times 2 to the power n plus 1, so that is one way of tiling so, so probably that kind of says that why chi f is upper bounded by 2 to the power n plus 1, because you can always I mean given any assignment to 0’s and 1’s you can always find a tiling that partitions it into two to the power n plus one rectangles.

Soon yes so that is what I was coming to next so, what is the what does the matrix looks like for the equality function what is m of f if f is the EQ function the identity matrix, so it is you have 1, 1, 1, 1, 1, 1 and you have 0’s everywhere else. So, what is the best way in which you can partition this into monochromatic rectangles? What is the best way? So, basically even if I have suppose let us say I have ok so let me put it in a different way?

Can I have a monochromatic rectangle having two 1's, I cannot write because the moment I try to capture two 1's in a monochromatic rectangle it will also include two 0s at least. So, in other words for each one, I will have a different monochromatic rectangle so what it means is that if I have this as the matrix for I mean this as my matrix the total number of monochromatic rectangles.

So, chi of EQ is actually, so from the first argument it implies that it is greater than 2 to the power n but it is also equal to 2 to the power n because there are exactly 2 to the power n 1s. So, each of these 1's, I will just set them as one matrix, no the one matrix for the rest is not true no but by definition a tile is basically a subset of zero one to the point, what do you mean by break it up into two halves you can always break it like this or like this, you cannot break it like this.

So, but we have, so that is what so you can either break it vertically or horizontally not diagonally having different values, so how many tiles will this have so each of these will be one so you have one for this and then one for this so the total number of tiles will be I think 2 to the power n plus 1, which implies that c of EQ is greater than n plus one but so basically the point is that after you pick some function the question boils down to, how best can you tile the matrix corresponding to that function?

Because once you have a tiling you have a lower bound so that is what this problem boils down and one is by construction how other way I mean how else can you get it this is the optimal value no, so that is what we have to show that anything lower than this is not possible so clearly it is lower bounded by 2 to the power n but why cannot something between 2 to the power n and 2 to the power n plus 1 exists.

So, if EQ is greater than n it can be greater than n plus 1 also this is giving me a stronger result that can happen, now there I did not say equal we said it is greater than n by the fooling set method. So, yes maybe you can think about it I mean there are many ways maybe you can argue that suppose if you take two diagonal elements in two adjacent rows having 0’s they cannot be put in one tile I mean in one rectangle.

Because if you put let us say these two in one rectangle then the 1s will also come in so these two must be in separate similarly these two must be in separate you can maybe formulate some argument that you need at least n other rectangles, so what are you doing so you start with, so let us take a small case, so let us say we have 00, 01, 10 and 11, so what do you do? So, that is what so I mean I do not think it is a difficult argument.

It is some but you have to I mean so that is if you want to give an exact answer you can do something like this so you think about it so basically it will be some random matrix I mean the matrix will have random values and basically it will depend on the inputs but for example this will be 0, 1, 0,1 it will be like a chess board so I am so I am not going to answer that so you think about it so everything should not be discussed you can reshuffle it.

Reshuffling is not a problem no but even without if you get a monochromatic tiling by reshuffling you will also get a monochromatic tiling by not reshuffling because of this definition because let us say that I am reshuffling these two rows, so that these two become contiguous so then instead of picking these two I can pick this one and this one prior to reshuffling, because these two are different entries.

So, if in one row you have two 1’s it means that x is equal to some x prime and x is also equal to x double prime can that happen so we will stop here I was also planning to do a review of at least all the classes that we have seen but clearly we do not have time for that but anyway so I think this is fine generating a problem with specific communication I mean you can always I mean you can always create a matrix which has a certain number of minimum monochromatic tiling.

So, let us say I want to generate a function which whose communication complexity is k so I will somehow define f function such that the minimum number of tiling will take 2 to the power k many rectangles, so again that is a combinatorial problem and then you argue that since it is the minimum is 2 to the power k the communication complexity is at least. So, another thing that I will just leave as a small exercise is again it is I think we discussed it.

So, it is no longer an exercise but nevertheless I will state it that this is a stronger method than the fooling set method.

(Refer Slide Time: 43:30)

So, in other words if so if f has a fooling set of size m then yeah chi of f is greater than or equal to m, so what this will imply is that if you can get a lower bound by the fooling set method then that will imply a lower bound by the tiling method and this is easy to prove. No I mean how do you frame that as a communication complexity protocol, the tiling, tiling is sort of the same so what that proof uses is this decision tree kind of a thing.

So, each time you sort and one comparison will divide into two yeah so it has a similar flavour to this approach that is true, so thank you very much.