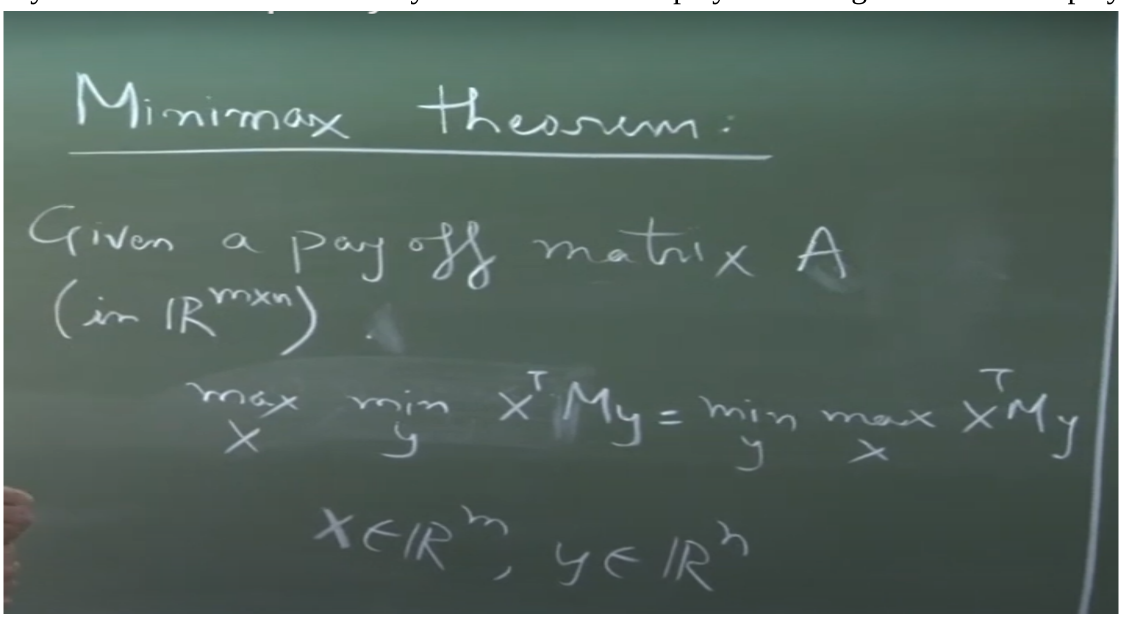
Linear Programming and its Applications to Computer Science

Prof. Rajat Mittal
Department of Computer Science and Engineering
Indian Institute Of Technology, Kanpur

Lecture – 34
Deterministic Communication Complexity

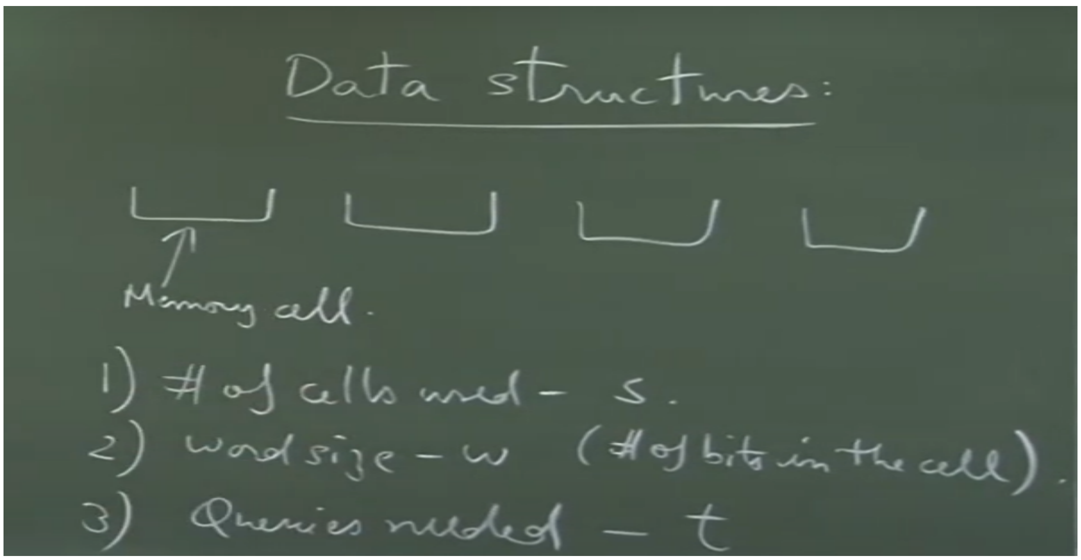
Welcome to another lecture of linear programming. We have been looking at Minimax theorem and just to put you in the context again, remember the idea was based on a 2 player 0 sum game. That means 2 players were playing that strategies in this case how many player strategies did player A have? M right row player had M strategies and column player had N strategies and we assume they can even play probabilistically right. So X is a distribution over all possible strategies of the row player. Y is a distribution over all possible strategies of the column player. So, when M gives the payoff we can say that there is the money which column player will give the row player.



In case, so M i j is the amount of money player B will give player A if A plays I B plays j right and then this is kind of the payoff if the if else plays sorry row player plays with X column player plays with Y right. Then we saw this maximin theorem it basically says that the that there exist a point the point is this X comma Y where this minmax will become equal right. At this point there is no incentive for the column player to switch the strategies there is no incentive for the row player to switch strategies to and this is called the Beautiful mind Nash equilibrium right. This is called the Nash equilibrium in this case right.

This shows that Nash equilibrium also always exist in 2 player 0 sum games and it is a direct in some sense corollary of what strong duality right. Today we are going to see another application of this nice theorem and it will take us places. So, you will see how important these ideas are in very very different applications right. So, I will start with data structures right. Why do we use data structures? How many people have done a course on data structures? You do not have to raise hands I know everyone has done it right rhetorical questions.

So, yes why do we use data structures? Very nice right. So, it is a format so that we can quickly run some algorithm or quickly run some task right it efficiently efficiently is a very nice important word we want to efficiently store our data. So, that is some questions some algorithms can be answered quickly on that structure data right. In a very simple model I am not do not worry about complicated data structures, but a very simple way in which I can think of data structures is that I am storing my data in memory cells. So, this is one memory cell and every query I can fetch one of these cells right.

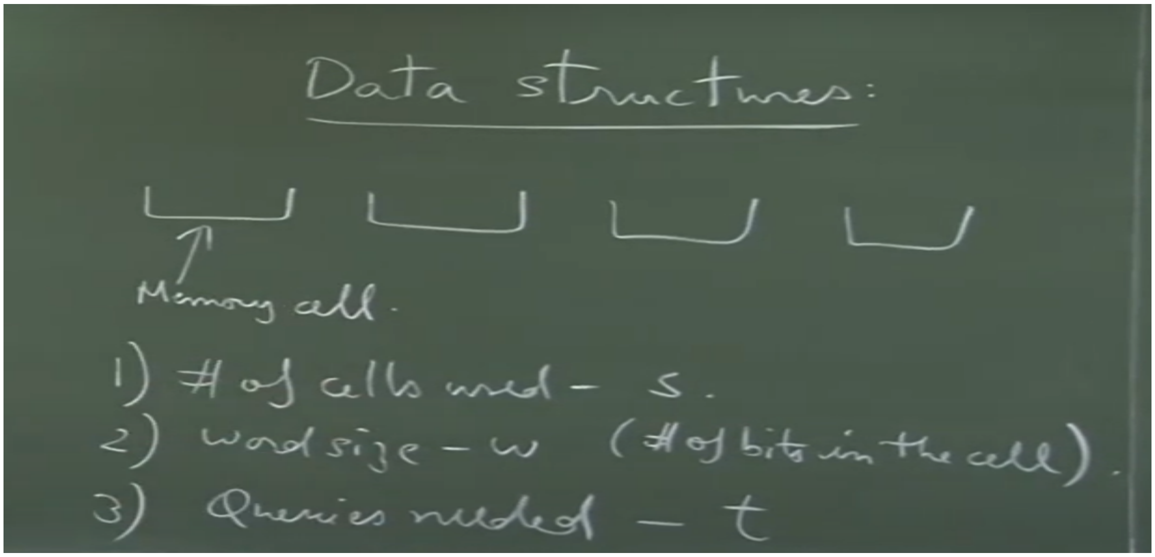


If this is the case then in some sense the efficiency is decide by three things. One is number of cells used let us call it S. Second I can say word size or W what is it? It is the number of what should be the word size intuitively in the cell right. It is the in the cell and generally data structures are optimized for a particular kind of question or an algorithm right. So, if I have a question in mind what other thing I want to what other thing I want to optimize time right.

Time in this case would be number of memory cells I will fetch right. So, I will just call it query is needed right and not generally it is query is needed in the worst case for the worst possible input right and this I will call T. And now depending on the problem you might have different kind of data structures such that these things will take different values. But generally the idea would be to say that you know if S is very small then W or T should be high. If these are high these are small then S should be some mixture of that should be high right.

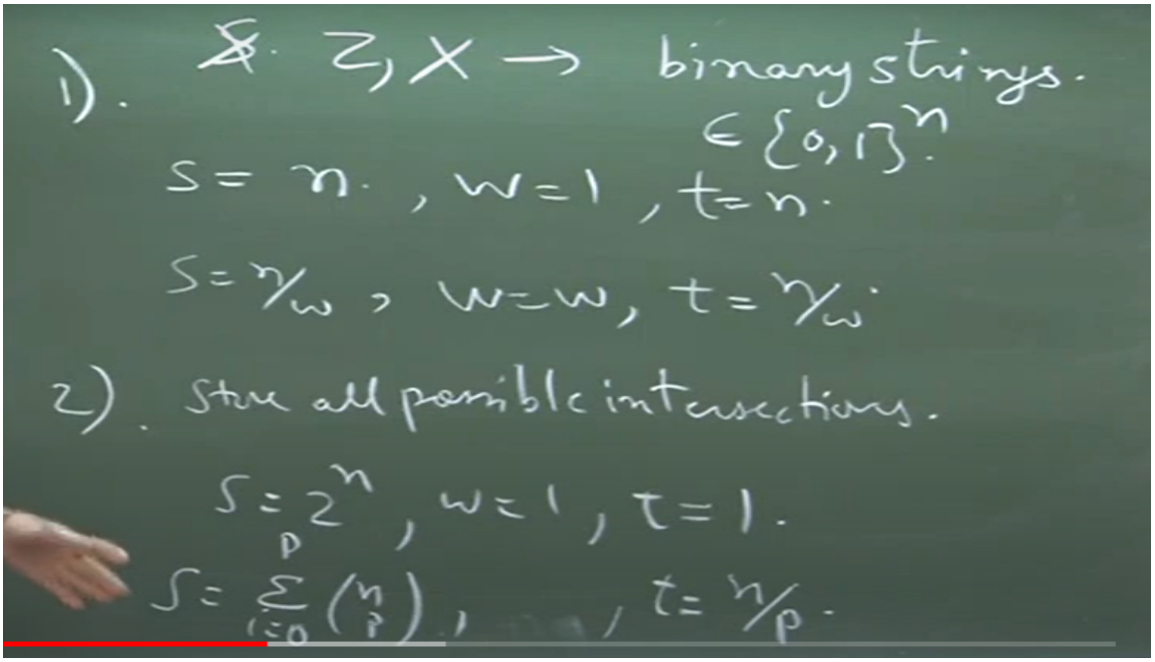
That is the kind of thing we want to prove as a complex theory. You want to say it is not possible to do it you know in just with one memory cell with one bit or something and you will always answer efficiently. You want to say that some task is hard some task is difficult if some combination of this thing is always going to be high right. And this will stay at the intuitive level, but this will make sense. So, let us try let us take an example right.

And my favorite example suppose IIT Kanpur has a master list of cheaters people who habitually cheat right. So, this is the master list with IITK right. And they want to encode it keep it in some data structures and every time one of us instructors find whoever has cheated you find we want to find out if those cheaters one of them is a habitual offender in that master list. The question is clear. And now since IIT Kanpur students cheat a lot this has to be done lot of times right every course consistently is trying to.



So, that means I need to store this efficiently I need to store it. So, that quickly I can recover this jokes apart what am I trying to do mathematically what operation I am performing mathematically no remember this is a master list I find I have some list of students who have cheated I want to find out if there is a person in my list who is also present here. This is set intersection and you can see that in spite of this made up example there are various cases where set intersection is an important problem right. So, data structure wise the problem is a set is stored already right. So, you want to store a set or let us call it y because x is generally the input or it is hard to cross out x z this is store a list store a set and then once the input comes I need to answer whether this is equal to 5 or not right.

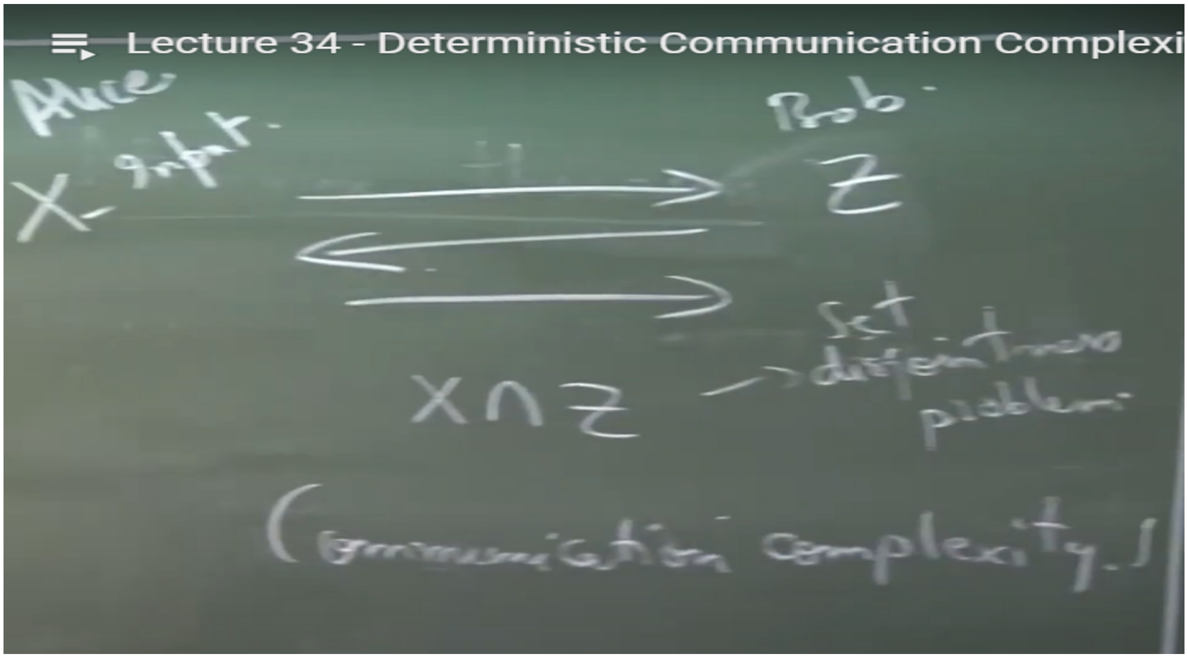
And you can see that this has many many applications right. So, now if you want to store how do you want to store it and again remember these are the important criteria's right what could be the possible ways to store it. Suppose the number of students the all the universe size is known to you n number of students. So, what could be the possible strategies how can you store it? That is not very what the randomize case let us very about deterministic. So, let us start with the simplistic right the simplest 1 is you have the set S right and sorry set Z and X as binary strings right in 1 to the n clear.



And say for this simple representation what is my S W T. what is S? S is n right, but I can say that let us if S is n you are saying W to be 1 and T is n right in the worst case correct. What I can do is I can even say that I am storing this as words of length W and W could be a parameter in that case W is W. Right now suppose I was only interested in T is there a better way to store I am do not care about S let us say array of bits of what does what do bits represent? That is what exactly this was right this was exactly what that was. So, now suppose I only care about T I have lot of space what would be your choice of representation very nice very nice what I can do is I can store all possible intersections. Correct and in that case T will be 1 right and not to bailable the point, but you can also say that I will store all possible intersections of sets of size less than some size P.

That will decrease S a bit that will increase T slightly. So, there is a trade of where you can reduce S a bit, but T will increase a bit right. So, because every set Z can be represented at with at most n by P sets and I just check the intersection with all those n by P sets. So, then this could be sorry and T would go like n by P like that that is the number of sets in which I can divide less than P right. So, all this again this is not the main intention, but what I am trying to say is that there are various strategies and there is lot of trade of between these quantities, but you also see that if I try to decrease 1 quantity other quantity will increase.

And this is true you can show such kind of results, but think about how hard it is because there are so many parameters and every time there are so many strategies right. You can set you can get intersection and if you want you want hash functions you want B S T is what you want to show is whatever thing you do whatever you have learnt in your data structure scores you cannot do better than some critical threshold right. So, you see how hard this problem is right, but if you want to give a lower bound you want to capture something which is intrinsic to this problem right. What do I mean by intrinsic to this problem? One way to notice that would be to say that you know let us say X is the input with the party L S and this data structure is with a with Bob Y sorry data structure was Z right should remind me right. So, data structure is with Bob and now I do not know what Bob is doing I do not care whatever Bob does I want to make sure I want to count the number of bits they communicate between each other such that at the end of it I get X intersection Z.

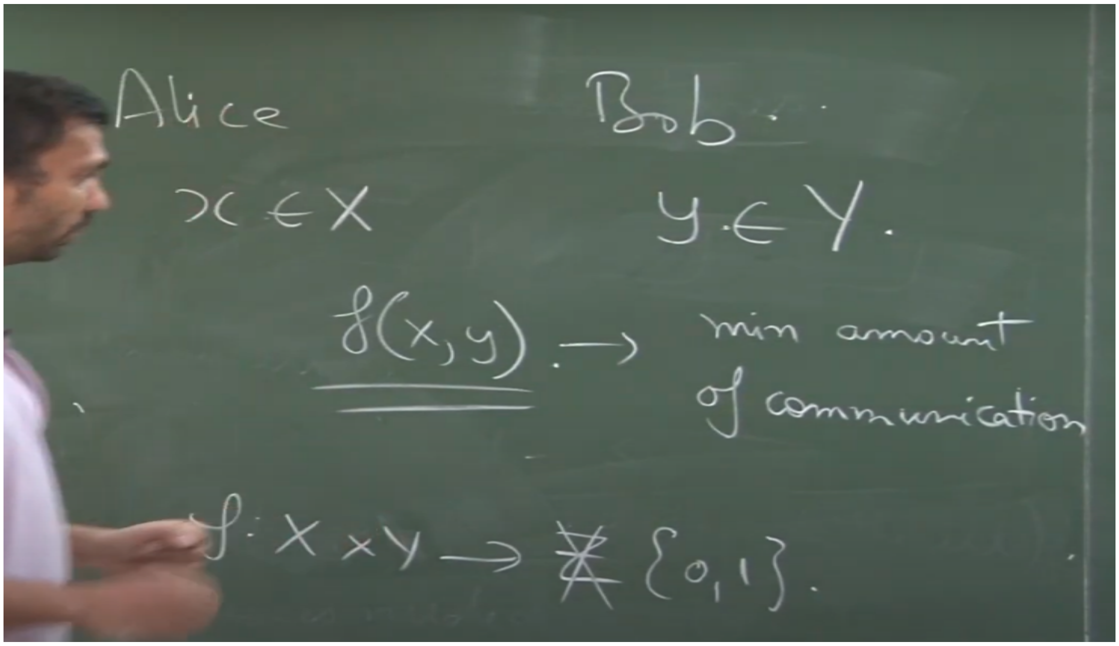


This is the idea again right it turns out that this is a very successful model this is called Communication complexity how many people have heard of it. So, this is called Communication complexity and this is not just useful in data structures you can give via circuit lower bounds you can give data streaming lower bounds you can definitely give like that good bounds on this kind of a thing actually comes from studying these kind of problems. And this is what we call set disjointness problem which is canonical difficult problem for Communication complexity. So, what we want to see today is how to give lower bounds in Communication complexity using max theorems right using linear programming. So, a very natural problem in data structures where it seems it is very hard to give lower bounds we will see actually that those lower bounds come up by first abstracting out what is difficult in this.

We are trying to say that however you store the data some bits need to be transferred and that means at least one of these things will be higher. So, this is the first abstraction which took a lot of time to develop and was a major insight I am giving it in 5 minutes, but definitely it is a huge deal right. So, first it turns out that to give lower bounds in this model you can reduce it to lower bounds in Communication complexity and not just in data structures, but in streaming, but in circuits in many cases lower bounds reduce to Communication complexity. This is the first thing I have only given an intuition for it, but now we will see how to actually give bounds in Communication complexity that would be more rigorous more robust. But is this kind of a informal idea clear how you can give lower bounds in data structures by giving lower bounds in Communication complexity.

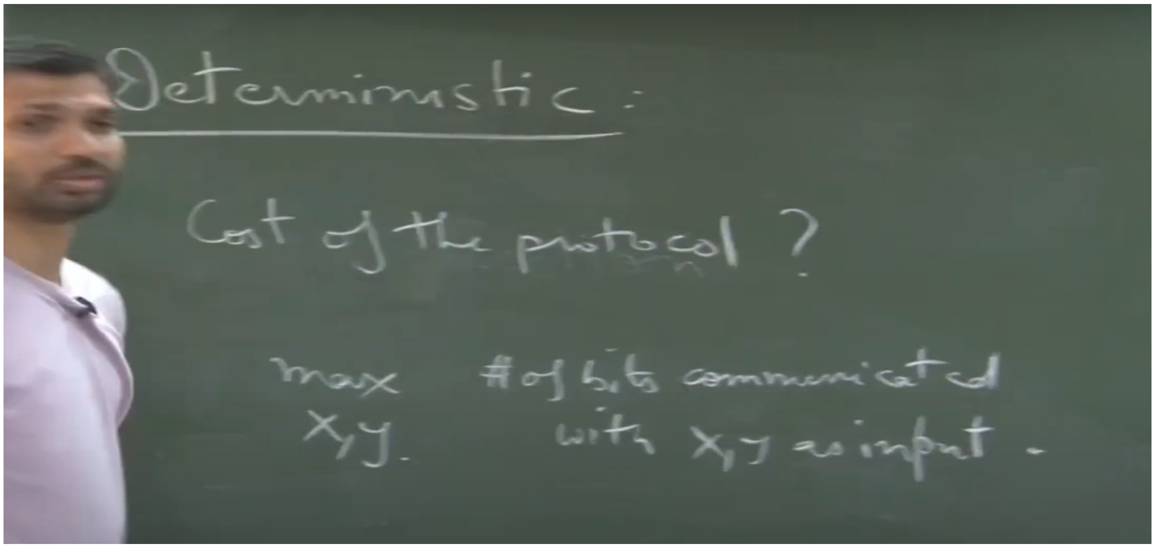
Communication complexity obviously has two parts giving nice protocols and giving nice lower bounds, but for these kinds of applications lower bounds are more important. But if you are mathematically inclined the protocols are amazing you should read about some protocols for Communication complexity how smartly this solves problems it is lot of fun. Though for today we are interested in lower bounds sorry. I will generalize the problem and tell you the generalized problem and then you see that you have match up. The general problem is I have two parties Alice and Bob.

Alice gets an input x Bob gets an input y and in the background I have a function in my which gives me a value and let us say whatever z z or actually let us simplify 0. I have a problem in mind Alice gets the input from x Bob gets the input from y and their task is to calculate f of x y using I do not care what they do here right that is I assume that they are all powerful. They can store y whatever way they want they can compute on y whatever things they want I do not care about it. The only thing which I want to minimize is how many bits are transferred between these two parties right. For example, remember in this case I can say they compute all possible intersection with all the sets and keep it around right even that was fine with me I never said how expensive it is to you know calculate the intersection.



So, to capture all these kind of data structures I am here saying I assume these are all powerful. The inherent difficulty in this task is that some part of information of x should get transferred to y or some part of information of y should get transferred to x if they want to calculate f x y. And since they want to calculate f x y they will have to create some amount of communication. If you give lower bound here it will give you a lower bound for this kind of task. So now, what is f x y in this case? X intersection set this is a set of all subsets of n this is a set of all subsets of n power set of n and this is saying whether there is an intersection between these things.

You mean to say in terms of computationally. So, let us say they have turing machine of infinite power like turing machine of infinite power that is good enough. So, now this is the model how do we give a lower bound on that? And even in this case there are deterministic strategies and there are randomized strategies. And we will see how to even give lower bound in the randomized case. But what does it mean for the protocol to be correct? There are many inputs in the case when I have a fixed protocol in mind.



What do I mean by the cost of the protocol? Because you know it might happen then on some x and y my protocol takes only sends only one bit of communication for some x and y it sends 100 bits of communication. So, what should be my measure of how many bits am I exchanging? Worst case right like we talked about here today right it has to be worst case. So, generally for deterministic protocols I say whatever this worst case is the most important measure which we study max over x y bits communicated with x comma y inverse.