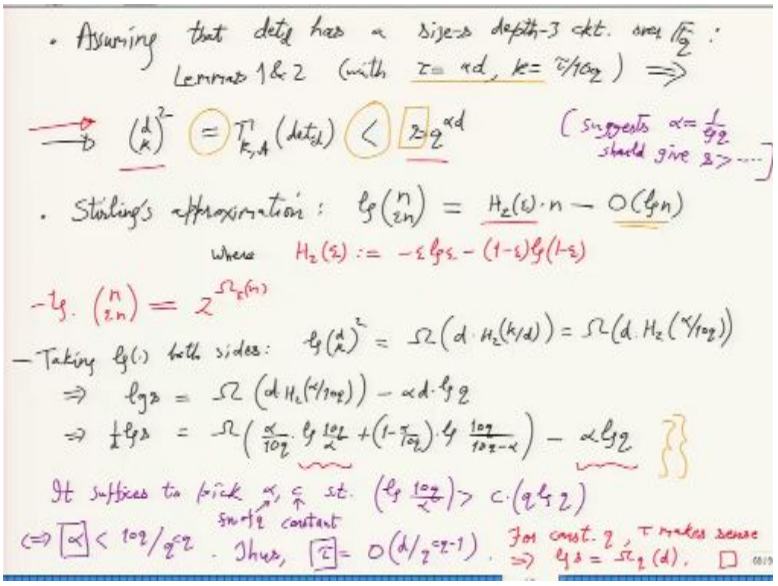
**Arithmetic Circuit Complexity**
**Prof. Nitin Saxena**
**Department of Computer Science and Engineering**
**Indian Institute of Technology-Kanpur**

**Lecture - 15**

,Last time we had shown these two properties.

**(Refer Slide Time: 00:18)**




That for the measure $\Gamma_{K,A}(det_d)$ is exactly $(\frac{d}{k})^2$ and measure of determinant is if you assume the determinant has a depth three circuit then it will be at most $sq^{\alpha d}$,. All we have to do is now pick a k and $\alpha$ so that you get a exponential lower bound on s. We want to isolate s here. For that we just simply have to do some calculation using Stirling's approximation.

$log\ (\frac{n}{\varepsilon n})$ is essentially a multiple of n where the multiplier is entropy function. And for example $(\frac{n}{\varepsilon n})$ from this $H_2(\varepsilon) := -\varepsilon log\varepsilon - (1-\varepsilon)log(1-\varepsilon)$. That gives you $2^{\Omega_\varepsilon(n)}$ where constants will depend on $\varepsilon$,. If $\varepsilon$ is for example half, then this is coming out to be around $2^{n/2}$ ,infact $2^n$.

$2^n$ and then the exponent the minus error term is very small. It is something like $2^n$ the whole thing divided by $\sqrt{n}$, The error term is only (½)log n, which is also what we have mentioned here. Yeah, but this is not true for all $\varepsilon$. Obviously, if you go far

away from 1/2 towards 0 or towards 1 then you will be in a problem. Then $\binom{n}{\varepsilon n}$ will not be so large.

That is why we have to do this calculation to fix things,.Let us take log both sides. So that will give you, on the LHS you will get log $\binom{d}{k}^2$ which is so using that formula, you get $dH_2(k/d)$ You are saying $\binom{d}{k}$ . This is the equation inequality. $\Omega$ d times entropy which is $\Omega(dH_2(k/d))$ is yeah, so that we have to read it from here.

The k / d is depends on $\tau$ /d which is $\alpha$ . $\alpha$ /10 q. We suggest that $\alpha$ will depend on q, which is why we assume q to be constant. And we get that log s is, again from that equation. From here we get that $log\ s = \Omega(dH_2(\alpha/10q)) - \alpha d\ logq$ . We can take d common. You get what do you get? You get $H_2$ . So the entropy is then use the definition.

Entropy will come out to be $\alpha$ / 10 q that is
$\Omega(\alpha/10q\ .\ log(10q/\alpha) + (1-\alpha/10q)log(10q/10q-\alpha)) - \alpha\ log\ q$ ,. Many things here are actually unimportant. What is important just if we can make sure that this part so think of this as the main part. If this main part is large enough then the main part will be able to compensate for this error term which is $(-\alpha\ log\ q)$ . We just have to compare these two.

Because the remaining part is it is already a positive contribution. We do not care about the positive contribution. You just care about this main positive canceling with the negative. So that is all. We will just pick, it suffices to pick $\alpha$, c such that
log (10 q / $\alpha$) >c $\alpha$ log q,. Then the main part will be a multiple of some constant times $\alpha$ log q, which will give you no sorry not $\alpha$ , c q log q this.

This is if this is more than q log q then you get yeah you get $\alpha$ / 10 q multiplied by this will give you c $\alpha$ log q, which you pick bigger than the negative error term. Is that clear? You just pick constants well, constant c at least and $\alpha$ is something that is

really dependent on q once we have fixed the constant c. This is actually an absolute constant and this is a function of q.

You will see how the dependence on q is right. This is inequality holds if and only if $\alpha < 10q/q^{cq}$. It is equivalent in equality. And so that fixes your $\alpha$ as a function of q only. And once $\alpha$ is fixed, that fixes your $\tau$ and once $\tau$ is fixed it fixes your k,. All the parameters are fixed. That fixes your $\tau$. That this means in the end is this part here the RHS here with these calculations this essentially becomes a positive constant.

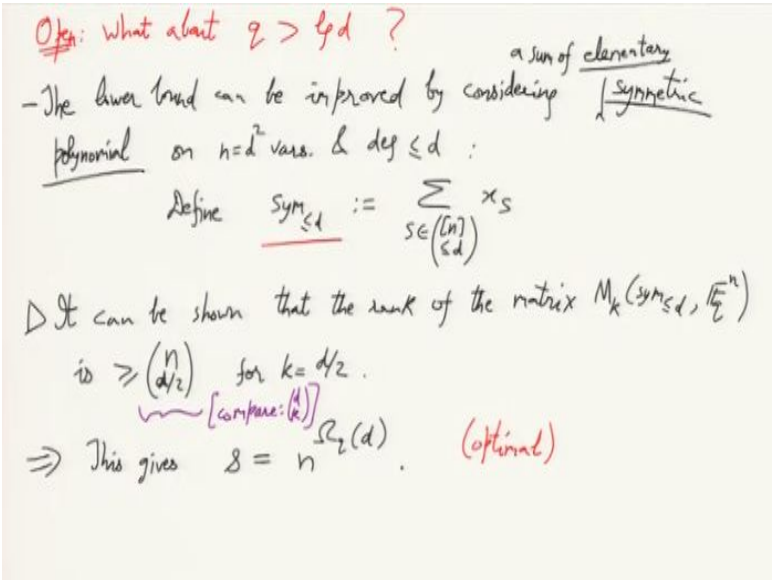
So log s /d is more than a positive constant which means that s is more than $2^d$, $2^{\Omega d}$. For constant q, $\tau$ make sense. What does what do I mean by make sense. Remember that our proof used $\tau$ to divide into cases whether the rank is less than $\tau$ more than $\tau$. We are using the fact there that $\tau$ is some integer, some integer bound.

In case q is very large, since you are dividing $d/q^q$. If $q^q$ for example is d or more than d then $\tau$ here is a fraction. That will not give you anything. You can think this proof will work as long as $q^q$ is sufficiently smaller than d. Certainly if q is $\sqrt{logd}$ the proof is fine or when q is just below log d /log (log d). That is the limit of this proof.

And this $\tau$ then appears also as constants in the $\Omega$, the lower bound constants are actually dependent on $\tau$. That actually is the main is the formal reason. log s is kind of dependent on $\tau$. If $\tau$ is very small then you do not get any lower bound. In otherwise you will just get that log s = $\Omega_q(d)$.That finishes the proof. You get s = $2^d$, $2^{\Omega d}$. Constants depend on q.

And this work up works up to q smaller than log d /log (log d). Is this clear. It is an open question to make it work for larger q's, larger finite fields.

**(Refer Slide Time: 12:16)**




If the field is really growing with the degree parameter then what do you do? Then is it possible that determinant or permanent d $\times$ d can be expressed as a sub exponential depth three circuit. That we do not know. Other remark is, is this minor optimization that why are we getting $2^d$, why not $d^d$ lower bound, which will be the optimal lower bound.

Obviously, we cannot get that for permanent because it is false. For determinant we do not know. But we can change the polynomial, we can look at some other polynomial and then it can be shown. The lower bound can be improved by considering a sum of, well not sum just elementary symmetric polynomial. And it will be elementary. If you look at elementary symmetric polynomials then the lower bound can be optimized.

This is a polynomial on $d^2$ variables like determinant and like determinant deg $\leq$d. But its definition is very different. It is actually it is a sum yes, less than equal to d. It is defined as so I should say sum. It is a sum because we are looking at all the monomials, not just for subsets of size d, but also d – 1, d – 2. It is a sum of these first d at most d degree.

Or the first d symmetric polynomials, we are just summing up. It is in homogeneous. It is a bit different from determinant and the definition will give you depth three

circuit of size, we are already $n^d$ ,. So $n^d$ is kind of d to d, $d^{\Omega d}$ is by definition. And that can also be shown as a lower bound by the previous proof. It is optimal of the matrix $M_k(sym_{\leq d}, F_q{}^n)$

This rank is $\geq \binom{n}{d/2}$. If you take k =d /2. Basically the rank lower bound argument in the previous proof can be improved. The rank upper bound argument on depth three circuits remains the same and then you match the two and you will get this lower bound. You will get the lower bound of $n^d$ So that is optimal.

**"*Professor - student conversation starts"***

***Student*** :And in this case how you are getting q d?

***Professor***: No the proof is the same. Rest of the thing remains the same. Calculations are all the same. ***"Professor - student conversation ends".***

Just the in the case of determinant we could only show a rank lower bound of this $\binom{d}{k}^2$ , which was which is anyways bounded by $2^d$ or $2^{\Omega d}$.

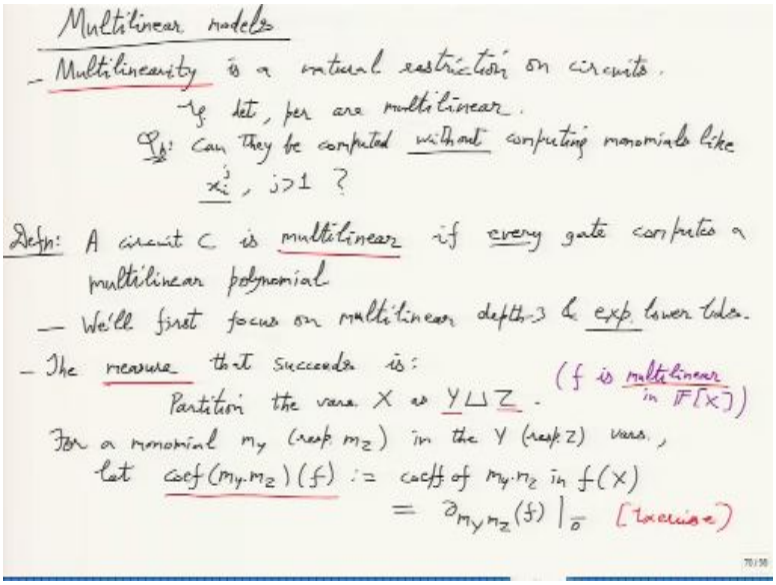
That was the fundamental limitation. Now we go slightly above $2^d$. I guess one syntactical difference, it may be an important difference is that there we were getting in the lower bound $\binom{d}{k}$ where d was the degree. Here we are getting $\binom{n}{d}$, where n is the number of variables which is $d^2$ . That $d^2$ actually gives you the kick.

Because this simple combinatorial estimate will be n/ $d^d$ . That will become $d^2 / d^d$ , that is $d^d$ . So that is where the improvement is coming from. Compare it with $\binom{d}{k}$ , n is $d^2$ . This is much better. Any questions?, Then we will move on to the next model and the next lower bound,. The technique will not change drastically.

It will be a smooth generalization of what we just saw, which is look at the matrix of derivatives at a point. And in fact at several points. The matrix of derivatives was the derivative was being evaluated at many points. That definition we will slightly

change. We we will continue to look at derivatives, but in a slightly different way. Matrix will be slightly different and the depth of the circuit we will now start looking at more depths, not just depth three.

**(Refer Slide Time: 19:52)**




Restriction will be multilinear, multilinearity. Multilinear models. What is that? Yeah. Your circuit ultimately will compute a multilinear polynomial which means that individual degree of every variable is at most 1. But moreover every gate will do that. Yeah, but for today, we will not look at the gates. That second thing is not very important. Just think of multilinear polynomials for now, for today's lecture. Multilinearity is a natural restriction on circuits.

Our most fundamental polynomials like determinant and then permanent, symmetric polynomials etc. they are all multilinear. You can ask the question that if the final polynomial is multilinear would it help if we used non multilinear monomials inside the circuit,. Maybe you can get a circuit that does not use $x^2$ for example at all. So can they be computed without computing monomials like squareful monomials like $x_i^j, j > 1$

So can we do without this $x_i^2$ $x_i^3$ ? Because ultimately they you have a feeling that these things will anyways get cancelled. How can their incorporation in the circuit help in computing determinant? So is this intuition correct? What do you think about

the simple-minded intuition? Can squareful monomials help you in computation? Yeah, but then mod computation has to be done. That is even harder.

How, there are too many. No ultimately you have to then eliminate mod gate.How do you eliminate mod gate by addition multiplication? Yeah, so you are making bold conjectures. For now all we know is our first algorithm for determinant was using sum of powers and Newton identities. Although there was no reason there was no monomial in the determinant which is squareful where the individual degree is more than one.

But still we started actually with the sum of powers inherently against multilinear computation. And that helped in the final circuit. That so every gate is actually working with these non multilinear monomials. It is absolutely not clear how to remove them. Then you may think about the second algorithm which is ABP the Mahajan, Vinay thing and there are also you would see that this close the closed walks.

They keep repeating with the variables with the non multilinear monomials and there is no clue how to eliminate that,.This intuition may be plainly wrong but well the question is well formulated, we continue. We try to understand why for what models will nonmultilinear monomials help us. And for polynomials, A circuit C is multilinear if every gate computes a multilinear polynomial,.

We will gradually prove bigger and bigger results. We will start with the depth three model with this restriction. Multilinear depth three. And then we will look at more general models. From depth three we will move to constant depth. From constant depth we will move to formulas. We will first focus on multilinear depth three and exponential lower bounds.

The thing about multi linear model is and why this has been studied so much is this partial derivative idea works very well in under this restriction. We will continue with the measures that use partial derivatives, in fact this partial derivative space. The

measure that succeeds here is a twist on what we did in the previous proof in the sense that it will not just look at the space of all the derivatives.

That may not be able to that may not be refined enough to give you a result. It will actually look at specific derivatives and in particular it will we will partition the variables into two parts depending on what kind of a circuit we are looking at. And the derivatives will be then the matrix of derivatives will be rows indexed by monomials in one part of the variables and columns indexed by monomials in the other part of the variables,.

That is a departure from the previous definition. It is a more refined measure than just look at all the derivatives. Partition the variables $X$ *as* $Y \sqcup Z$. For a monomial $m_Y$ or respectively $m_Z$ in the Y variables respectively Z let coefficient ( $m_Y$ . $m_Z$ ) in a polynomial f, f has variable set X. What is this? This is just what it says.Note that $m_Y$ times $m_Z$ remains a multilinear monomial because Y and Z are disjoint.

We are just asking for the coefficient of this monomial in f. It may still be zero. But it is not always zero. This is somewhere maybe we write down that f is multilinear in over a field variables x, multilinear polynomial. Coefficient this coefficient in f this is also equal to. The coefficients are actually in general also they are related somehow to the derivatives.

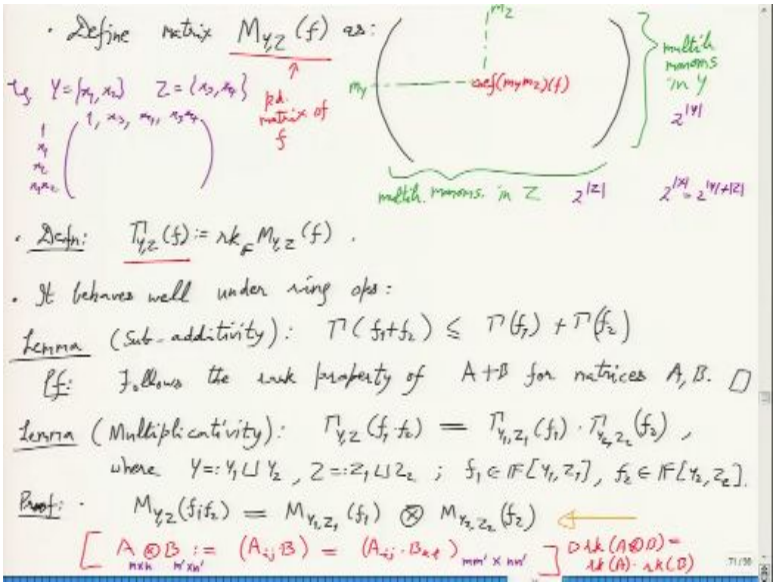
This is also you differentiate f, is this enough? Yeah, so problem with this derivative is yeah, if you look at the polynomial $yz^2$ let us say y and z are just single variable. If you look at $yz^2$ and you differentiate by y differentiate by differentiate by y z then you will get 2 z ?But that is not a coefficient. In fact in the polynomial $yz^2$ the coefficient of y z is 0.

There is a slight gap between derivatives and coefficient and that gap is filled by saying that derivative at the point 0. You differentiate by this monomial and then set all the variables to 0. That is what gives you the respective the corresponding

coefficient. You can see this simple exercise particularly for the multilinear case, otherwise you have to generalize this slightly.

Do you have to generalize it or is it always, it is always true? Yes, this it is a short proof for this, right? Instead of saying this derivative at 0, we will just say coefficient of f. It is the same thing. And then we have a matrix whose rank is will be the measure.

**(Refer Slide Time: 32:26)**




Define matrix $M_{Y,Z}(f)$ as monomial $m_Y$ is here, $m_Z$ is here and use this coefficient or extract coefficient $m_Y$ times $m_Z$ coefficient of the monomial $m_Y$ times $m_Z$ in f. These are all the monomials in Y. In fact, sorry not Y, Z; multilinear monomials. We only care about them and these are all the multilinear monomials in Y. If the degree of f is d, then you just look at multilinear monomials in Y or Z of degree up to d.

That is the, it is a finite matrix. Entries are either zero or nonzero constants from the field, field f. Many of the next lectures these methods will be field independent. I will not talk much about the field,. Now, from now on many of the methods are actually field independent methods.

**"Professor - student conversation starts"** Every entry which is corresponding to some Y and some Z,. This Y and Z is a partition. Yeah Y are the Y variables Z are the

Z variables. X is the universal variable set. Because there are many YZ's. In that. No Y, big Y is the variable set let us say the left variables and Z big Z is the set of, is another variable set disjoint from big Y. **"Professor - student conversation ends".**

Let us call this Z the right variables. And you look at all the multilinear monomials in the left variables versus all in the Z variables and then draw the matrix where entry will be of the product corresponding to the product extract the coefficient. Like if you may have $\{y_1, y_2\}$ or if you use x variables we can write in terms of that $Y = \{x_1, x_2\}$ and $Z = \{x_3, x_4\}$ and then you have here $1,\ x_1,\ x_2,\ x_1x_2$.

And there you have $1,\ x_3,\ x_4,\ x_3x_4$ It is a $4 \times 4$ matrix and yeah you do not have any other monomial possible. You have 16. In general you will have here that many. Yeah, it actually it is not really dependent on the degree. You just use everything. $2^{|Y|}$ many and $2^{|Z|}$. Overall you have $2^{|Y|+|Z|}$. $2^{|X|} = 2^{|Y|+|Z|}$.

That is what the, it is basically that. We are looking at all the multilinear monomials in the end, but how we will define draw the matrix, that will obviously change the rank and the measure. What is Y and Z? That will change everything actually. It will be hard work and later on very hard work to identify the Y and Z. It will use a lot of probability theorems done probabilistically.

Sometimes this method will just probably fail. There will be no partition. It is not a general method, which is why we have put all the restrictions and we will put more or other restrictions. Yeah, it will work for constant depth multilinear circuits and multilinear formulas. Yeah, it is a, that is a surprise these are major results.. Once we have that definition, we also get the $\Gamma_{Y,Z}(f) = rk_F(M_{Y,Z}(f))$. This is the rank of the matrix over the field. And this matrix is also called partial derivative matrix of f which is an incomplete name.

Because it is not just all the partial derivatives, but how they are arranged. Partial derivative matrix of it with respect to a given partition. And its rank is what the

measure is. As experienced before, this measure will have nice properties. When you have $f_1 + f_2$ and $f_1 f_2$ , the measure will behave well. It behaves well under ring operations. Unsurprisingly it has sub additivity, I suppress y and z now.

Let them be fixed. That is $\Gamma(f_1 + f_2) \leq \Gamma(f_1) + \Gamma(f_2)$ How do you show this?The coefficient operator is linear and this is obviously, inherited from the derivative operator being linear. You have two matrices one each for $f_1$ , $f_2$ and then use, follows the rank property of A + B for matrices A, B.
Rank of ( A + B ) is at most rank of( A) + rank of (B) , just that.

And you know that this is the best you can say. It can be strictly smaller. It is not additivity but sub additivity. And second is multiplicativity. Yes, here we want to, this will be tricky. Here we want to look $\Gamma(f_1 f_2)$. But we will have some assumptions which is that? $f_1 f_2$ have disjoint variables. Since $f_1$ , $f_2$ fhave disjoint variables you can think of each of these variables sets having their own partitions.

There are actually four variable sets. In those terms we will write. $\Gamma_{Y,Z}(f_1 f_2) = \Gamma_{Y_1,Z_1}(f_1) + \Gamma_{Y_2,Z_2}(f_2)$ and there will be an equality where $Y = Y_1 \sqcup Y_2$ $Z = Z_1 \sqcup Z_2$ And $f_1 \varepsilon F[Y_1, Z_1]$; $f_2 \varepsilon F[y_2, z_2]$ ,. Those are the assumptions. So $f_1$ is basically on the variables $Y_1 \cup Z_1$ . It has its own measure, its own matrix. And $f_2$ is on $Y_2 \cup Z_2$ variables. It has its own matrix.

And then when you multiply them then you will take essentially the row variables of $f_1$ row variables of $f_2$ union and the same thing with columns. Is the statement clear? What is the proof of this? This is strong, this is an, this is a perfect equality. Already, multilinearity definition of a circuit is motivated by this. This is the property that will demand that every addition gate or the addition gate is not important.

The multiplication gates, they should all be computing multilinear. Hence, the input to a multiplication gate should be disjoint variable.. Let us define the matrices and then well so $M_{Y,Z}(f_1 f_2)$ first matrix. Second matrix is $M_{Y_1 Z_1}(f_1)$ and third matrix is

$M_{Y_2,Z_2}(f_2)$, who will guess the relationship between these three? Oh, we have read the survey. I you look at the rows in $M_{Y_1}$ and $M_{Y_2}$, the rows are in completely different variables.

And the rows, why they are all the monomials in $Y_1 \cup Y_2$ .There is this construction in matrix called tensor product. That will perfectly match this index set. If you take the matrix product you will get, at least in terms of the row count this matches. The number of rows in $M_{Y_1}$ and the number of rows in $M_{Y_2}$ their product is the number of rows in $M_Y$ And columns, column count also matches.

We can potentially ask the question whether these two things are equal,. So is there anyone who does not know the definition of tensor product? Yeah, I think, this sometime, we did see that sometime. Oh, that long ago. In general tensor product of matrices is defined as, this really does not need any assumptions on the dimensions of the matrices.

Take any matrix A and any matrix B they may not even be square matrices. And basically, inside the matrix A you have these entries $A_{ij}$ 's. You just replace $A_{ij}$ by a matrix which is B scaled up. You put B and you multiply every entry there by $A_{ij}$ ,. This is further this further expands as $A_{ij} \cdot B_{kl}$ and then do this for all i,j, k,l.

But the way you will organize the matrix is given in the middle, which is it is the A matrix with every entry replaced by a bigger matrix given by B. Let us write the dimensions. You have m $\times$ n, $m' \times n'$. Then the final number of rows is $mm' \times nn'$. Yeah, this is a weird construction. You may be ready to accept addition of matrices and multiplication of matrices.

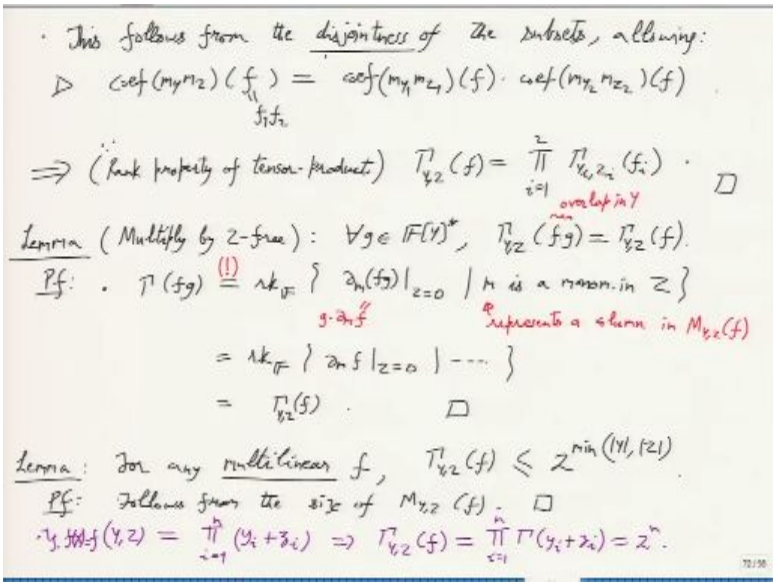
But may reject this definition of tensor product.The simplest way to motivate this is when you look at a polynomial ring with one variable and go to two variables, what happens?, If you think about this for a day then you will come up with this definition. It is basically at the level of algebra, you are just increasing the number of variables,.

That construction is it is yeah it is a very fundamental construction and it is called the tensor product.

Here explicitly, it looks like this at the level of matrices. What we have is, this question is then well defined. You take these two matrices, take the tensor product and compare it with the matrix for $f_1$ times $f_2$. Now how do you show that these two are equal? You are looking at a coefficient of $f_1$ and a coefficient of $f_2$. And in the tensor product you multiplied. But since they come from everything is disjoint.

This coefficient also appears then in $M_{Y,Z}(f_1 f_2)$ and moreover when you multiply $f_1$ with $f_2$ the polynomial multiplication does not involve convolution. You are just picking one coefficient of $f_1$ one from $f_2$ and the product is a new is a final coefficient. There is no convolution happening. So use all these assumptions.

**(Refer Slide Time: 49:56)**




This follows from disjointness of the subsets which allows

coef $(m_Y m_Z)(f_1 f_2) = coef\,(m_{Y_1} m_{Z_1})\ (f_1) \cdot coef\,(m_{Y_2} m_{Z_2})\ (f_2)$. There is no small. No, maybe I have jumped a step. In any polynomial f multilinear polynomial f you can factorize it like this. This is correct. And f here we will take $f_1$ $f_2$ Now when you look at the first extraction, this $Y_1$ $Z_1$ extraction from $f_1$ $f_2$

Yeah this will just extract the product monomial from $f_1$ the coefficient from $f_1$ because there can be no contribution from $f_2$. So $f_2$ can only contribute a constant term. Constant of $f_1$ $f_2$ is the only, is a thing which is common in all these equations. We have this property and this property implies from the rank property of tensor matrices.Using the rank property of what is the rank of A $\otimes$ B?

Yeah how do you show that? Yes, that is a simple exercise. Using the definition you can show that the rank ( A $\otimes$ B )= rank (A) rank (B)., in the end it is a simple construction. It the rank just multiplies. To prove these things as an exercise then. But what if you do not know what Eigen values are? Rank is just rank is visible, right? Eigen value is something very hidden.

Yeah, no easiest is just use rank by first principles. Yeah, I mean you use using the definition of rank, you can say that after a point, the columns are just zero or something., then you go up to $r_1$ columns in A and let us say $r_2$ rows in B. Everything else you set to zero and then you show that now for this chunk the rank will be $r_1$ $r_2$.

Yeah, so I think by first principles it is easiest and you do not need any assumptions.. Using this coefficient extraction of f, $f_1$ times $f_2$, you get the tensor product and from the tensor product now you get the measure which will be $\Gamma_{Y,Z}(f) = \Pi \ \Gamma_{Y_i,Z_i}(f_i)$ . We did this for product of two polynomials but it is also true for any product. Same proof.

Measure of f is equal to measure of product is equal to product of measures. Is that clear?, We can do one more property, or yeah, then we can leave. They are all just basic properties of the measure independent of the model. This is multiplication by Z-free. So for any g nonzero that is Z-free, which means it has only Y variables,. Remember that the measure is defined with respect to y cross well Y in the rows Z variables in the column.

If you have a polynomial that is Z free then the measure for it is just single column,. So if you multiply by such a polynomial, then what happens? If you multiply f which is an arbitrary multilinear polynomial, you should multiply it by g and then look at the measure. Yeah, it does not change. That is the claim. Exactly. Measure does not change if you multiply by Z-free g or you multiply by a Z-free g symmetrically.

Yeah, it is again simple, but we have to still be careful about this. $\Gamma(fg) = rk_F \{ \partial_m(f_Y|_{Z=0} \mid m \text{ is a monomial in } Z$ .. It is a new claim. This actually follows m is the column in your matrix. What does the what does a column in your matrix represent? That is the question.

It basically represents that we have differentiated f with respect to m and then gotten a polynomial which is Z-free, it is only in Y and its coefficients are put in the column, right? Think in those terms. This is this represents a column in the matrix. The rank of the columns is exactly equal to the rank of the resulting derivatives. And at the point 0 Z = zero. Y is not to set to 0.

We are actually looking at polynomials in Y; Z has been set to 0. Is this clear? So this also just think about this. It follows from the definition of the matrix. Once we have this, what can you say about $\partial_m(fg)$; g does not have any Z. So g comes out. This lemma yeah it does not it is not subsumed in the previous case. In the previous case everywhere we needed we used disjointedness. There is something else.

In particular, as they are saying f and g have, I am not assuming that f g is a multilinear product. This is actually you guys should write it down overlap in Y. So g has Y variables f also has Y variable. Both have both of them may have $Y_1$ and that will be giving you $Y_1^2$. **"Professor - student conversation starts"** You are not partitioning Y, you are only partitioning Z and it also works.

Yeah, either way, you have to prove something else. This is giving the proof. You have to prove something else, which is this. **"Professor - student conversation**

**ends".** So g is a nonzero polynomial, we have taken it out, so then we can just drop it which is nothing but $\Gamma_{Y,Z}(f)$. Yeah, g you can just drop this. Yeah and one last thing is any questions about this.

Last property will be upper bound on the rank, trivial upper bound. For any, this is where we are assuming multilinearity actually first time. For any multilinear $\Gamma_{Y,Z}(f) \leq 2^{min(|Y|,|Z|)}$ Is that clear? Just either minimum number of well either number of columns or number of rows whatever is smaller is an upper bound.

This just follows from the size of $M_{Y,Z}(f)$ matrix. But since this is so lazily done, do you think it is optimal? What is the example? thanks to Ramprasad. f( X) you take sorry no, that is not a good idea. I want to use my notation. f ( X) is you take these X to be 2n variables and these 2n variables, first n variables are these $y_i$ variables. Second, the later n variables are $z_i$ variables. And look at this product,.

What is the measure for f? Now use the lemmas. This tensor product thing tells you that it is yeah it will boil down to basically $\Gamma(y_i + z_i)$ and the measure of this is clearly 2. You get $2^n$ which is the upper bound. It matches..