

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

**Lecture - 09**

So we have to finish last steps of the depth-reduction, log-depth reduction.

(Refer Slide Time: 00:21)

Theorem (Valiant, Skyum, Berkowitz, Rackoff '83):  
 Let degree polynomial  $f$  have size- $s$  circuit  $C(\pi)$ . Then, there is a poly( $sd$ )-size,  $O(\log d)$ -depth circuit  $C'$  computing  $f$ .

Proof: • Why assume that  $C$  has  $f_{\max} \leq 2$  & that  $C$  is right-heavy  
 i.e.  $\forall$  gate  $v$ ,  $\deg(v_L) \leq \deg(v_R)$ .  $C, f$  are homogeneous.  
 left child right child

• By  $[v]$  we denote the polynomial computed at gate  $v$ .  
 Also,  $[v]$  will be made a node/gate in the new circuit  $C'$ .

• Defn: For gates  $u, v$  define gate quotient  $[u:v]$  as:

- $[u:u] = 1$
- For a leaf  $u$  &  $u \neq v$ ,  $[u:v] = 0$
- $[u_1 + u_2 : v] = [u_1:v] + [u_2:v]$
- $[u_1 \times u_2 : v] = [u_1:v] \times [u_2:v]$

$\triangleright \deg([u:v]) \leq \deg u - \deg v$ .  
 $\triangleright v$  does not occur in  $\text{tree}(u) \Rightarrow [u:v] = 0$ . Pf: Base case is that of a leaf.  $\square$

So maybe we first look at the theorem statement. Theorem statement is this VSBR theorem which says that if you have a degree  $d$  polynomial with circuit complexity  $s$  then there is a  $\log d$  depth circuit as well computing it and the size will increase nominally by  $\text{poly}(sd)$  or  $2^{\text{poly}(sd)}$ . So this is of interest only when the degree is not too high.

If the degree is exponential then this is not an interesting theorem, but when the degree is let us say comparable to the size in that case this is proving something very strong. It is saying that in  $\log$  depth similar size, you can compute the same polynomial exactly. Some of the details of this are put in the assignment.

(Refer Slide Time: 01:26)

• Intuition behind  $[u:v]$  : assumes  $v$  on the right-side.  
 Say,  $[u] = A[v] + B$  for some polys  $A, B$ .  
 We would like to talk about the circuit that computes  $A$ .  
 This is formally obtained by  $[u:v]$ . ( $= A$ )

• Defn: • The frontier at  $\deg m$  is  

$$\mathcal{F}_m := \{v \mid \deg v_L \leq \deg v_R < m \leq \deg v\}$$

•  $\mathcal{F}_m$  are the deepest multiplication gates with  $\deg \geq m$ .

$\Delta$   $u \neq v \in \mathcal{F}_m \Rightarrow [u:v] = 0$   
 Pf:  $v \notin \text{tree}(u)$ .  $\square$

Lemma (Frontier expan.)  $\therefore \deg u \geq m \Rightarrow [u] = \sum_{w \in \mathcal{F}_m} [u:w] \times [w]$ .

• If  $\deg u \geq m > \deg v \Rightarrow [u:v] = \sum_{w \in \mathcal{F}_m} [u:w] \times [w:v]$ .

Pf: • We'll do reverse-induction of  $\text{depth}(u)$ .  
 • Base case:  $u$  is the deepest, i.e.  $u \in \mathcal{F}_m$ .  
 $\Rightarrow \text{RHS} = \sum_w [u:w] \times [w] = [u:u] \times [u] + \sum_{u \neq w \in \mathcal{F}_m} [u:w] \times [w] = [u]$ .

And the main technique is frontier expansion lemma assuming homogeneous circuits and the frontier expansion lemma is this. It says that polynomial computed at a  $[u]$  can be expanded in terms of frontier nodes where frontier nodes are, according I mean they are below  $u$  and that is the first place where the degree jumped. Let us say it crossed  $m$  parameter. So there is one for  $u$  and one for the quotient gate.

So we will use these two things actually to expand  $[u]$  you need the quotient gate and then to expand the quotient gate you will need two quotient gates in a summand. Here you only need one quotient gate but to expand that you have to use in every summand both of them will be quotient gate. We have to deal with them simultaneously.

So in the case when you expand  $[u]$  you need the degree of  $[u]$  to be more than the parameter, degree parameter and in the quotient case, you need the parameter to be in between.

**(Refer Slide Time: 02:52)**

• Case  $u = u_1 + u_2$  :  $[u] = [u_1] + [u_2]$   $\deg u = \deg u_1 = \deg u_2$

$$\begin{aligned}
 \text{RHS} &= \sum_{w \in \mathcal{F}_m} (u; w) \times (w) = \sum_{w \in \mathcal{F}_m} (u_1 + u_2; w) \times (w) \\
 &= \sum_{w \in \mathcal{F}_m} (u_1; w) \times (w) + \sum_{w \in \mathcal{F}_m} (u_2; w) \times (w) = [u_1] + [u_2] \\
 &= [u]
 \end{aligned}$$

•  $[u; v] = [u_1; v] + [u_2; v] = \sum_{w \in \mathcal{F}_m} (u_1; w) (w; v) + \sum_{w \in \mathcal{F}_m} (u_2; w) (w; v)$   
 $= \sum_{w \in \mathcal{F}_m} (u; w) (w; v)$

• Case  $u = u_1 \times u_2$  : In the non-base-case  $u \notin \mathcal{F}_m$ , so  $\deg u_2 \geq m$ . homogeneity C

$$\begin{aligned}
 \Rightarrow [u] &= [u_1] \times [u_2] = [u_1] \times \sum_{w \in \mathcal{F}_m} (u_2; w) \times (w) = \sum_{w \in \mathcal{F}_m} (u_1; u_2; w) \times (w) \\
 \bullet [u; v] &= [u_1] \times [u_2; v] = [u_1] \times \sum_{w \in \mathcal{F}_m} (u_2; w) (w; v) = \sum_{w \in \mathcal{F}_m} (u; w) (w; v)
 \end{aligned}$$

□

The proof was by induction. This is just induction on the size of the circuit. Or you can also think in terms of the depth, induction on the depth. So addition, multiplication are the only two cases that we dealt with. And here we saw that we needed a homogeneity assumption. That was important. And we also needed these assumptions here. So actually, for the addition gate it is important that you have homogeneity assumption.

Otherwise the degree difference can be too large. If it is too large, then no matter which frontier gate you pick, it will miss one of the summands. It will, you would not be able to apply frontier expansion lemma provably, it will probably fail. So you need the degrees to be very close, so in fact in this case degrees will be equal.

And then, depending on the frontiers, you can get information either side and the sum will give you everything. This seems pretty important for the proof to work and some assumptions also needed in the product.

**(Refer Slide Time: 04:05)**

- We'll use this to write the depth-reduced circuit  $\leq$ !  
We'll take a top-down approach.
- We'll recursively compute  $[u]$ ,  $[u:v]$  from nodes in  $C$  of a lower degree.
- Let  $\exists w := \exists_m$  for  $m := \deg(u)/2 > 1$ .  
Now,  $[u] = \sum_{w \in \exists(u)} [u:w] \times [w] = \sum_{w \in \exists(u)} [u:w] \times [w_L] \times [w_R]$   

[  $[u]$  is an addition gate with fanin  $< 2n$   
 its input mult. gate has fanin  $= 3$   
 $\deg$  of  $[u:w], [w_L], [w_R] \leq \deg(u)/2$  . ]
- $\exists(u,v) := \exists_m$  for  $m := \deg(uv)/2 > 1$ .  
Now,  $[u:v] = \sum_{w \in \exists(u,v)} [u:w] [w:v] = \sum_w [u:w] \times [w_L] \times [w_R:v]$
- $\deg(w_L)$  could be  $> \deg(u,v)/2$ .
- We apply frontier expansion lemma again:

That was for the frontier expansion lemma. Now to apply this lemma to reduce the depth, we will pick  $m$  optimally to be half. So if you want to expand  $[u]$  with respect to frontier gate  $w$ , then you want  $w$  to be such that to be the node where the degree just crossed half of the degree of  $u$ . And we said that this is the deepest. So deepest things are the frontier of half.

I am not sure because I think ultimately, everything can cancel out and give you zero. So that monotonically is a dangerous assumption. We do not want that. But maybe that can be dealt with. So if it is zero, then that whole circuit can be removed, sub circuit. . Maybe monotonicity can be also assumed. That is possible, because we are only after an existential.

Well, we want the existence of a small circuit. So either way it will work . So then there was this technical point which is to do with the way we are proving this step production that if you expand  $u$  with respect to this half frontier, half degree frontier, you will get multiplication fanin 3,. So you apply frontier expansion lemma once.

And then you apply, you use the fact that  $w$  since the degree is growing there the first time it has to be a multiplication gate. So you get left and right children and you get  $u$  expressed as a sum of things which are all product gates with fanin 3. But this is not

enough. So we cannot stop here. The problem was that although the degree of  $u, w$  is good, small enough, well for  $u$  it is fine.

The degree of all three is less than  $\deg[u/2]$ , but this is using the quotient gate. So later you also have to apply frontier expansion lemma on the quotient gate and then what you will get is with the problem is you get this. Here also currently the multiplication fanin is 3 but  $w_L$  is something whose degree is not half of the output. So output is  $u/v$  but its degree may be exceeding the  $(\deg[u] - \deg[v])/2$ , right?

Because the way we use these frontiers their parameter is between the degree of  $u$  and  $v$ . It is the mean actually. That tells you that does not tell you much about  $w_L$  gates degree. So what was the solution for this? Expand again. We expand again and then the degree, then the multiplication fanin becomes  $2 + 3$ , it becomes 5. And that is where we will settle.

(Refer Slide Time: 07:45)

$$\Rightarrow [u:v] = \sum_{\substack{w \in \mathcal{F}(u,v) \\ p \in \mathcal{F}(w_L)}} [u:w] \cdot [w_L:p] \cdot [p] \cdot [p] \cdot [w_L:v]$$

$$\triangleright \deg[u:w] \leq \deg u - \frac{\deg(uv)}{2} \leq \frac{\deg(uv)}{2}$$

$$\triangleright \deg[w_L:v] \leq \frac{\deg(uv)}{2} - \deg v \leq \frac{\deg(uv)}{2}$$

$$\triangleright \deg[w_L:p], [p], [p] \leq \frac{\deg(uv)}{2}$$

$$\text{Pf: } \leq \frac{\deg(w_L)}{2} \leq \frac{\deg(uv)}{2} \quad \square$$

• Eventually, we reach a case where  $\deg[u]$  or  $\deg[u:v]$  is at most 2. There we can explicitly compute in  $\Sigma\Pi\Sigma$  (depth 4).

• Each appln. of "frontier expns." halves the degree  $\Rightarrow O(\log d)$ -depth in the end.

• The size in the end is  $\text{poly}(2d)$ . [Exercise]

• The final circuit  $C'$  has alternating  $+$ ,  $\times$  gates & fanin  $\times \leq 5$ .  $\square$

So we will settle here. Here the frontier gates will be, we will have to do with the  $w_L$ . And that will guarantee the degrees to be all of them to be half of the output which is  $u/v$ . So these inequalities you can check, they are easy to believe. But this is the, this is the rationale for multiplication fanin being 5. 5 is a strange number to have. But we get 5 because of this because of this double application of frontier expansion lemma for quotient gates.

So that is the structure theorem. It is not only a depth reduction result, but it also tells you something pretty strong about the structure of the circuit. Let us continue from here. Eventually we reach our case where  $\deg[u]$  or  $\deg[u:v]$  is at most 2. The degree will become after these, these iterations. So iteration means that you write  $u$  or  $[u : v]$  as a sum of multiplication gates or fanin 5.

That is one you think of it as one step. This step you will repeat many times, and at some point degree of these inputs to the multiplication gates will fall to a constant say  $< 2$ , or  $\leq 2$ . So when that happens then this case you solve trivially. These we can explicitly compute in depth 2, so in  $\Sigma\Pi$ , or maybe even  $\Sigma\Pi\Sigma$ .

That will really depend, but definitely  $\leq \Sigma^3$  depth 3 will be enough for this trivial computation or brute force computation in the end and how many iterations are there before you reach this constant degree phase. Invariant is that degree of the output is always halving in this one step. So degrees always halving. So then if you started, if at the root the degree was  $d$  then this has only  $\log d$  iterations.

So the invariant is that each application of frontier expansion, which is not once but twice if required. So each application of this type of frontier expansions half the degree which means invariant means that you get  $\log(d)$  depth. This is repeated squaring at the level of sum of products. Sum of products of 5 things. So depth is  $\log(d)$  and what is the size in this overall process?

With some care you can show that the size is not exploding five times or  $s$  times every time. Because the way we have written it, it seems that every time we are we have a big sum of products. But if it is big sum every time for every factor then ultimately you are growing up to  $s^{\log d}$ , right? In the worst case that could have happened,  $s^{\text{depth}}$ .

But that is not the case because these frontiers also keep on reducing. So you are actually the frontiers which you are using are first of all, they are all disjoint. Because in every next so in the next iteration the frontier is with a lower degree parameter. So it cannot be the same as the ones before. So they are actually falling. Whatever you see in the current subtree from there you pick the frontiers.

You can show that overall it is not too bad, just poly (sd). The size in the end is poly (sd). So this you can also do as an exercise for the algorithmic analysis in the assignment and as already remarked this not only reduces the depth but also makes the circuit very structured. So this the final circuit  $C'$  has alternating plus multiplication gates, . Because you are always using this  $\Sigma\Pi$  and inside also it is  $\Sigma\Pi$  and so on.

So they alternate and the fanin of the multiplication gates  $\leq 5$ . Any questions? So this is the complete structure theorem for arithmetic for algebraic circuits. It is a very powerful result and well unprecedented in a way because these things you do not see in Boolean circuits. And it is surprising because so if the multiplicative fanin is 5 and the depth is let us say exactly  $\log d$  then what is the circuit what is the maximum degree that the polynomial can, that the circuit can compute?

It is kind of  $d^{5^{\log d}}$ , which is  $d^{\log 5} = d^{\log 5}$ . So that seems to be the magical bound, which up to which if you go then maybe you can do some non-trivial computation. But there is no need to go beyond that. If your goal is to compute only a degree  $d$  polynomial, then there is no point going doing computations which exceed the degree bound  $d^{\log 5}$ .

So  $d^{\log 5}$  is something like  $d^2$ . So if you want to attain a polynomial of degree  $d$  then there is no point doing any computation where the degree intermediate degree blows up beyond this  $d^2$ . And this is, this is very intuitive but you never know in advance. I mean maybe it was possible to have intermediate computations which computes something very high degree.

And then in the end, the degree reduces, that might have helped. But this theorem is saying that it does not help. Just a moderate growth is all you require. So if there are no questions, then we will prove something even stronger, well stronger in some sense than this result.

(Refer Slide Time: 16:41)

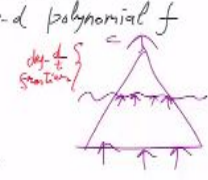
Reduction to bare minimum depth

- By the efficient  $O(d)$ -depth reduction we know that:  
To prove hardness results for a deg- $d$  polynomial  $f$  it suffices to study  $O(d)$ -depth.
- Now we'll reduce it, further, to depth-4:

Theorem [Agrawal-Vinay '08, Koilara '12, Tavenas '15]:  
Let  $f$  be a deg- $d$  poly. computed by a size- $s$  circuit.  
Then, for  $t \in [d]$ ,  $f$  has a homogeneous  $(\sum \pi^{O(d/t)}) (\sum \pi^t)$  circuit of top-fanin  $\frac{s(d/t)}{t}$  & size  $\frac{s(d/t)}{t}$ .

[  $\sum^k \pi^{d'} \sum \pi^t$  circuit looks like  $\sum_{i=1}^k \prod_{j=1}^{d'} f_{ij}$   
where  $k = \text{top-fanin}$ ;  $t = \text{bottom-fanin} \Rightarrow \deg f_{ij}$ ;  $f_{ij}$  is in dense representation ]

[ To optimize the size take  $t = \sqrt{d}$ ; giving  $k \approx \text{size} \approx \frac{s\sqrt{d}}{t}$  which is better than  $s^d$ . ]



Now we will reduce the depth to bare minimum and what is that? What is bare minimum? Some may think that even  $\log d$  is too much depth. Why not constant depth? Why not 100? So we will ultimately go to depth 3. So we will say that well depth 3 is also in a way enough, but no depth 3, I mean talk about the minimum right. So the bare minimum will be 3.

So if you go to depth 2 then you are just saying that you have a mathematical polynomial. It is just a sum of monomials. So there is nothing to do there computationally, but when you talk about depth 3 then maybe computationally you can come up with better identities and more compact representations and we will say that will ultimately show that if you can do something on a high depth scale, then you can also see something non trivial at this depth 3 scale.

It will not be really optimal as the previous result but still you will get a non-trivial depth 3 representation of the polynomial. So we will go to depth 3 in the end. That was



a big surprise. It happened not many years ago. Non trivial means well trivial is  $s^d$ . If the degree is  $d$  and number of variables is let us say, well it is always bounded by  $s$ . So trivial will be just write down all the monomials. So it will be around  $s^d$ .

That is trivial. So non trivial could be  $s^{\sqrt{d}}$ . That is very non trivial. So which means that if you actually have  $s^d$  potential, potentially many monomials but you have been able to compress that into  $s^{\sqrt{d}}$ . And those things will be visible, even a depth 3 scale. So the point of at least one point of all this is that if you since you have seen this efficient  $\log(d)$  depth reduction.

Now you know, now we know that to prove hardness, hardness results for a degree  $d$  polynomial  $f$ , it suffices to study  $\log(d)$  depth. So if you have a polynomial family in mind and you want to show that its circuit complexity is very high, then it will be enough to show that assuming that the depth of the circuit is  $\log(d)$ . This will be a perfect result.

Whatever you can show for  $\log(d)$  depth would apply also to other depths, computing that polynomial  $f$ . So why cannot we have that kind of a connection to I mean with constant depth circuits. So we will that is the thing that we will really optimize and we will optimize it extremely. We will go to depth 3 in the end. So for now let us so now we will reduce this further to, so as Abhibhav suggested first we settle with depth 4, Let us first show this for depth 4.

So why depth 4? What is it that you can do with depth 4, what cannot do with depth 3? Well, so one vague feeling is that if you have a circuit so this business is not about repeated squaring. This is something much simpler. So why cannot, why cannot you just cut your circuit into top and bottom parts? The inputs are here. The top part is basically computing these polynomials with the inputs that it gets here.

So that you can write as a  $\Sigma\Pi$  in  $\Sigma\Pi$  representation. So sum of products of variables and constants. And the same thing you can do with the bottom part. So then you have

$\Sigma\Pi$  representation on the top and many  $\Sigma\Pi$  polynomials in the bottom feeding into the top. So that is depth 4. So basically we call, so we call this flattening. So you flatten the top part and flatten the bottom part and flattening, this flattens the original circuit to depth 4.

So this is the, simple thing is the key idea actually. We just have to implement this. So if you implement this then you get the following theorem. Agrawal Vinay showed this just before 2008. Then later on it was improved. So it keep getting improved in the next decade. So the version that we will do is the following. Let  $f$  be a degree  $d$  polynomial computed by a size  $s$  circuit.

Then pick a parameter  $t$  and cut your circuit in such a way so that the top part or the bottom part computes degree  $t$  polynomials and the top part computes with respect to them degree  $(d/t)$ . So then for that will give you the result that  $f$  has a homogeneous  $\Sigma\Pi^{O(d/t)} \Sigma\Pi^t$ . That is the top part. Circuit of top fanin which is for  $\Sigma$ , top  $\Sigma$  that is  $s^{O(s/t)}$ . And overall size is okay.

So this is the version by Tavanias. You should think of  $t$  as  $\sqrt{d}$ . And then you see that any degree  $d$  polynomial with a size  $s$  circuit also has a  $\Sigma\Pi$ ,  $\Sigma\Pi$  representation where the multiplication fanins are  $\sqrt{d}$  and the size is  $s^{\sqrt{d}}$ . Now  $s^{\sqrt{d}}$  as already remarked this is a highly non trivial size because brute force would give you  $s^d$ .

So why is the  $d$  being reduced so heavily? So that is a non-trivial representation. Oh, Agrawal Vinay did not care about the exponent. They just gave  $s^{O(d)}$ . But they had some other I mean that proof has some other advantages, which I will not go into right now. That it has size  $s$  circuit. No, well homogeneity is kind of you can assume that you can think of  $f$  as broken into homogeneous parts.

No, so you think of  $f$  as a sum of homogeneous parts, and for each homogeneous part, you apply this theorem. You will get the same result in the end because, I mean this  $s$  and  $d$  actually ideally you should think of them as comparable and this  $s$  will change

to  $s$  times  $d$  with the homogeneous parts. So maybe I should formally define this notation.

No so top fanin is already written here and the two  $\pi$  are already defined. The bottom  $\Sigma$  is full fanin  $s^t$ . So this part is just you can think of this as computing  $t$  degree  $N$  variate polynomial. So this is fully expanded, gives you  $n$  to the  $t$  monomials. More interesting is the top part, this part. So here also the fanin of  $\Sigma$  is only  $s^{(d/t)}$ . Then it could have been anything, it could have been  $s^d$  in general.

So the existence of a small circuit for  $f$  has a meaning even for depth 4 representation,. So whatever magic you can do with more depth, which essentially means that whatever magic you can do at all using an algebraic circuit you can do in depth 4. You can what? This upper bound is kind of tight. So there are then subsequent papers that suggests that this cannot be improved.

Yes. So this part is  $d/t$  degree. So if the top if that much of part has already computed degree relative to the inputs, its inputs,  $d/t$  then the bottom part could only compute maximum degree  $t$ . So that gives you for the bottom part it gives you  $\Pi^t$ ,  $\Pi$  with fanin  $t$  and the top part gives you  $\pi$  with fanin  $d/t$ . So that will be the basic implementation. Using the structure theorem we saw previously.

So that will be used, this is not independent. And moreover this is a homogeneous depth 4 representation which means that well which does not mean anything for product gate but for the  $\Sigma$  gates maybe  $f$  also has to be homogeneous. Otherwise this is impossible. So I have to assume that also that  $f$  is a homogeneous polynomial.

So for an arbitrary  $f$  think of a homogeneous part and if that homogeneous part has a small circuit then you can convert it into a depth 4 representation which will be homogeneous which basically means that whenever you are adding up in one of these two  $\Sigma$  gates, you always add up polynomials of the same degree. If  $f$  is inhomogeneous then this top  $\Sigma$  this will also be inhomogeneous.

So actually the top  $\Sigma$  will be inhomogeneous but the bottom  $\Sigma$  will not be, it will remain, it will be homogeneous. So instead of saying that I just let us just assume  $f$  to be a homogeneous polynomial computed by some circuit homogeneous or inhomogeneous, it does not matter. It can be converted to a homogeneous depth 4 where both the  $\Sigma$ , yes. I mean you just you will just look at the homogeneous parts.

For that this theorem is actually for that  $\Sigma^k \Pi^{d'} \Sigma \Pi^t$ , what is this model. So this circuit looks like it is a sum of product of polynomials where the sum adds  $k$  many summands and each summand is a multiplication gate with  $d'$  many factors. And each factor so this  $f_{ij}$  is given to you fully. So  $f_{ij}$  is a trivial representation. So where  $k$  is top fanin,  $k$  is called top fanin.

And  $t$  is called, you can call it bottom fanin and this will bound the degree of  $f_{ij}$ 's., so bottom fanin is this is basically a degree bound on these  $f_{ij}$ 's, these factors which appear in  $\Sigma \Pi f_{ij}$  representation;  $f_{ij}$ 's are given to you in full. Is in dense representation. So dense representation is just give all the monomials. No need to compress anything there.

And second point is that to optimize the size, just take  $s = \sqrt{d}$ . Because that is when  $t + d/t$  will be minimized and that will be  $2\sqrt{d}$ . Forget the 2 because there is a big  $O$ , so it will be  $s^{\sqrt{d}}$ . So giving  $k$  roughly equal to the size roughly equal to  $s^{\sqrt{d}}$  which is non trivial. So it is better than  $s^d$ . So that is the point of this theorem.

So now in terms of understanding the hardness of polynomials, the question boils down to looking at depth 4 representations of a very special type. So if you have a degree  $d$  polynomial then you want to express, so you want to study whether that polynomial can be expressed as  $\Sigma \Pi^{\sqrt{d}} \Sigma \Pi^{\sqrt{d}}$  non trivially. So it can obviously be represented in this way using  $s^d$  size.

But can you make it non trivial? Can you make it  $s^{\sqrt{d}}$  size representation. So if you show that you cannot that representation does not exist, then it means what? It means that no size  $s$  circuit exists. If this type of a depth 4 representation does not exist for your polynomial then actually you have shown that no circuit exists for the polynomial of size  $s$ . So you have a strong connection between general circuit lower bounds and very special type of circuit lower bounds. Any questions?

(Refer Slide Time: 36:25)

Proof: • We'll use Saptharishi (2016)'s version.

- Let  $C$  be the  $O(d)$ -depth circuit, of size  $s$ , computing  $f$ .  
Wlog, for each internal gate  $g \in C$  we've a homogeneous expr.  

$$g = \sum_{i \in [d]} g_{i1} \cdots g_{is} \quad \text{--- (1)}$$
 where  $\deg g_{ij} \leq \deg g/2$ . [double ind. poly( $s, d$ )-time.]  
 $\Rightarrow$  repeating eqn (1) gives a  $\Sigma\Pi\Sigma\Pi$ -exp of size  $s^d$ .  
 • To reach to  $\Sigma\Pi\Sigma\Pi^t$ , we'll incrementally open this up using  $\gamma_n(d)$ .  
 (i) For each summand  $g_{i1} \cdots g_{is}$ , with some  $g_{ij}$  of  $\deg > t$ , expand  $g_{ij}$  one step further (& same on  $g$ ).  
 (ii) Repeat this process till all  $g_{ij}$ 's (on RHS) have  $\deg \leq t$ .  
 • Each expr., like (1), grows the top fanin by a multiple of  $d$ . We intend to show that this happens  $\leq O(d/t)$  times!

So let us go into the proof of this, which is just the implementation of that. **“Professor - student conversation starts”** This depth so the depth reduction in 3 is it field dependent or field independent? **Professor:** It is field dependent. Or it is characteristic of the field dependent. **“Professor - student conversation ends”**. Does not care about the field per se. Just characteristic should be larger than  $d$ .

This  $d$  should be smaller than the characteristic. Otherwise or characteristic can be 0. But if you are looking at a prime characteristic smaller than  $d$ , then it gets into trouble. That is one way, but it does not work. So you will see the proof, it is a complicated idea. It will first increase the depth to 5 and then it will gradually reduce it to 3. You will see that, not today. Oh, you are saying that just flip the operators?

Oh that is blasphemous. No if you are given a sum of all the monomials you cannot express it as a product of linear polynomials.  $\Sigma\Pi$  flipping would mean that you are

expressing a sum of monomials as a product of linear polynomials right? It is impossible. I mean it is mathematically impossible, forget the size. For whatever size, it is impossible.

Since it is based, this result came after a sequence of paper, so we will just use Ramprasad's survey version. Instead of going into these papers and what did they do, we will just look at a really refined version subsuming all of them. So it will just start where we stopped in the previous result, the structural result that let  $C$  be the  $\log(d)$  depth circuit, computing  $f$ . So we directly start with log depth.

And we assume the size parameter to be the same which is  $s$ . So we renamed the size. This we can do by the previous structural result. Moreover, we can assume that for each internal gate  $g$  and  $C$  we have a homogeneous expression. Basically that this frontier expansion thing that we use  $g = \sum_{i=[s]} g_{i1} \cdots g_{is}$ ;  $i$  cannot exceed the size of the circuit, so it is 1 to  $s$  and so if  $g$  is a multiplication gate then you just have these five things being multiplied.

If  $g$  is an addition gate then you are adding up several products, each being a product of five things coming from the lower layer. So let us call this equation, equation 1. And what are the degree bounds? So degree of  $g_{ij}$  is what?  $\deg(g) / 2$ . That is the important thing. And maybe keep in mind that all this could be done in polynomial time but randomized. That is not important.

So whatever we will do hands on will also be doable in randomized polynomial time. So you can actually get the depth 4 circuit also in that much time to express. Yes, I think so yes, I think that can be assumed. So this equation 1 if you keep this throughout all these levels, that will also give you a  $\Sigma\Pi$  representation, I mean, this is just opening up the circuit. So root is computing  $\Sigma\Pi$ .

And then things below are also  $\Sigma\Pi$  and  $\Sigma\Pi$ . So you can keep on unfolding the circuit and ultimately, you will get a  $\Sigma\Pi$  representation. Or if you stop somewhere, just be

the leaf you get a  $s \Sigma \Pi \Sigma \Pi$  representation. What is the size of that? What does this trivial implementation give you. So repeating equation 1 gives the  $\Sigma \Pi \Sigma \Pi$  representation of size how much?

Well if you just keep on unfolding this then you will every unfolding would blow up the fanin of  $\Sigma$ , by something like  $s^5$ . So and you will do this  $\log(d)$  time. So this will give you something like  $s^{\log(d)}$ . I think more than  $s^{\log(d)}$  if you just did the trivial thing. Well this cannot be correct, we correct this. Each time you are, no no it will give you  $s^d$ .

So what happens is that if you write  $g_{i1}$  again as  $\Sigma \Pi$  and up to  $g_{i5}$   $\Sigma \Pi$  and multiply this to ultimately express it as  $\Sigma \Pi$ . You are also producing a lot more  $g_{ij}$  kind of polynomials. And you are unfolding each of them. So ultimately what you get is just a trivial representation of sum of monomials. So this will actually give you  $s^d$ . So you do not want to do this in all the levels.

If you do it in all the levels you will get the trivial result. So the idea here would be to stop at this threshold of  $t$ . So as soon as the degree of  $g_{ij}$  falls below  $t$  we stop this process. We do not apply it on  $g_{ij} \dots$ . So we will just truncate this process at a threshold of degree  $t$ . So to reach to  $\Sigma \Pi \Sigma \Pi^t$  so this actually would have given us  $d/2$ . Actually even in one step. It is already equation 1 is already of this type.

It is  $\Sigma^s \Pi \Sigma \Pi^{d/2}$ , because  $g$  if the  $g$  is the root then its disagree is  $d$ . So it is already of this size, this type. So to reduce this  $d/2$  to  $t$  we will incrementally open this up using equation 1. So if  $g_{i1}$  for example, if  $g_{i1}$  has degree more than  $t$  then we will use equation 1 on that. And so expand out  $g_{i1}$  as a sum of products and the degree will half. The degree of  $g_{i1}$  will half.

So if you keep doing this then at some point you will reach this threshold of  $t$ . But then as you are unfolding and multiplying, you are increasing the additive fanin. You are blowing it up, right. So we have to analyze this carefully. So let us do that. So for

each  $g$  summand  $g_{i1} \dots g_{ir}$  with some  $g_{ij}$  of  $\deg > t$ , expand  $g_{ij}$  one step further. And same on  $g$ , same transformation.

So when you are expanding  $g_{ij}$  obviously you are also getting a bigger fanin representation for  $g$  because  $g_{ij}$  is a factor in the summoned of  $g$  representation and so keep doing this. So again if  $g_{11}$  is of degree  $t$  then you expand it if degree will half in the summands. And if  $g_1$  at the same time if  $g_{12}$ 's degree is more than  $t$  then also apply it on that simultaneously.

But then the problem is that  $g_{11}$  and  $g_{12}$  they are, in the summand you have a product of those. So when you increase the fanin of  $g_{11}$  and  $g_{21}$  and you multiply these things out the overall top fanin blows up, it squares kind of. That is the price you pay in step 1. But after paying these, these costs in the end where do you get? You get to a point where till all  $g_{ij}$ 's on RHS have degree  $\leq t$ .

So in the end you have this original polynomial  $f$  computed at the root  $f$  equal to sum of products, where each factor in the product has degree  $\leq t$ . So this is by definition  $\Sigma \Pi \Sigma \Pi^t$  representation, is that clear? So now we have to see what is the size of this guy. Is it still  $s^d$  or something smaller? Is the process clear?

So the process does not need any proof, the process is just continue, they just repeat step 1. It will of course it has to halt. It cannot go on add infinitum because the degree is being reduced, in fact halved,. So it will stop at some point. So now we want to analyze what happened when it stops, right? How much of size blow up happened. So each expansion like 1 grows the top fanin by a multiple of  $s$  because top fanin in equation 1 is  $s$ .

So whenever you apply it on  $g_{ij}$  the fanin becomes  $s^2$ . So it grows by a multiple of  $s$ . We intend to show that this happens at most  $d/t$  times. That will be the crux of the proof, show that this blow up that is happening multiplicatively by  $s$  happens at most



$d / t$  times. So that would give you a bound of  $s^{d/2}$  instead of  $s^d$ . How do you show that? Why should you get  $d / t$ ?

So starting degree was around  $d$  or maybe  $d / 2$  and you want to reduce it to  $t$ , right? So how many iterations will it take?  $\log(d / t)$  but that is, that is a different analysis because that you are doing on, in parallel you are doing it also on other  $g_{ij}$ 's. So that does not give you it is not that it gives you  $s^{\log d}$ . So we have to do this carefully and show that this blowup is actually happening only  $d / t$  many times.

So we have to do this simultaneous analysis. That in a product gate, when we are doing this on all the inputs, what is the overall blowup.

(Refer Slide Time: 54:36)

• In eqn (1) if  $\deg g =: d'$ , then largest deg  $g_{ij}$  has  $\deg \geq d'/5$  (in every summand).  
 • Moreover, the 2nd largest deg factor has  $\deg \geq \frac{d' - d'/2}{5-1} = d'/8$ .  
 $\Rightarrow$  in each new summand there are two factors of  $\deg \geq d'/8$ .  
 $\Rightarrow$  Whenever we use eqn (1) to expand a  $\deg \geq t$  factor, we introduce at least one more factor of  $\deg \geq t/8$  (in each new summand).  
 • By homogeneity, there can be  $\leq \frac{8d'}{t}$  factors (in a summand) of  $\deg \geq t/8$ .  
 $\Rightarrow$  # iterations (in the process)  $\leq 8d'/t$ .  
 $\Rightarrow$  # summands in the end  $\leq 2^{O(4d'/t)}$ .  
 • Finally, the  $\deg \leq t$  factors can have at most  $n^{O(t)}$  many monomials.  
 $\Rightarrow$  C converts to  $\Sigma \Pi \Sigma \Pi^t$  circuit with top-fanin  $\leq O(4t)$  & size  $\leq 2^{(8d' + 4t)}$ .  $\square$

So in equation 1, if a degree of  $g$  is let us say, temporarily it is  $d'$  then largest degree  $g_{ij}$  in any summand has degree how much? What is the largest degree  $g_{ij}$ ? So upper bound you know is  $d'/2$  but what is the lower bound? So that you will get by homogeneity since the degree of the product is  $d'$  one of them or the largest one has to be more than  $d'/5$ , in every summand, summand  $i$ .

Otherwise there might be just some very bad summand is low degree. You will not have  $d'/5$  factor anyway. The whole summand may have degree, let us say 2. So I

mean, just think of a fixed  $i$ , when I am making these statements. But then  $i$  has to vary over all the summands. So whatever I will say has to be true for all the summands. But in your mind, you can fix  $i$  to be 1 just as think of the first summand.

So in the first summand largest degree factor has to be at least  $d'/5$  and it is at most  $d'/2$ . That you know. So moreover, the second largest degree, so what can you say about the second largest lower bound? So if you remove the largest degree  $g_{ij}$  then how much of the degree remains? At least half remains, So that is  $d' - d'/2$ . So this much remains and how many factors remain, 4 factors. So you get  $d'/8$ .

So the largest degree is at least  $d'/5$  and the next one is at least  $d'/8$ . So both of them in particular are at least  $d'/8$ . That is all I wanted. So it implies that in each new summand there are two factors of degree  $d'/8$ . So from this I will deduce that whenever we expand, we use equation 1 to expand a degree greater than equal to  $t$  factor we introduce at least one more factor.

So after applying after expanding by equation 1 in every summand, in every new summand, there is something still with degree  $(t/8)$  surviving. I do not remember why  $t/8$  is important. Then I could have done with  $(t/5)$ , not sure. But anyway, so the point is just that when you expand a high degree  $g$  then in the summands, in every summand there is some moderately high degree  $g_{ij}$  also present. Let us take it degree  $(t/8)$ . That is fine.

So even when you are simultaneously doing this on all these factors that have degree at least  $t$  you still end up with  $(t/8)$  degree factors and then I by homogeneity I want to claim that hence this operation you cannot do more than  $8d/t$  many times. So does that sound right? Let us check that. So by homogeneity there can be less than equal to  $(8d/t)$  factors in a summand of degree greater than equal to  $(t/8)$ .

So if you are looking at a factor of degree  $t/8$  then it can only have  $(8d/t)$  many factors overall. So this means that the number of iterations is bounded by  $(8d/t)$ . Is

this believable? That will give you that number of summands in the end is less than equal to  $s^{O(d/t)}$ . So you have let us say you start with the in the very beginning you have a summand with five factors and simultaneously you so all of them say had degree more than  $t$ .

So you simultaneously applied this expansion on these 5. So each of these will now produce summands of 5 factors. So if you look at the product of these respective factors then you have 10 factors now or you have 25 and so on. So every time these number of factors keep on going. But can the number of these iterations be more than this  $(8d / t)$ ?

**“Professor - student conversation starts”** Because each time you, each equation you double the number of factors of degree  $t / 8$ . Right. Is the doubling important or do you only need 1. It doubles every time right because we want a total of  $(8d / t)$ . So only if it doubles every time can we say that this will happen in. But no, I do not want the log. That log has a mistake. **“Professor - student conversation ends”**.

I think I need 2 because some, there is some reason which I do not remember. No the summand increase, the summand increase we are analyzing in the end, we do not worry about that. That is the last step. For now we just want to show that the number of times we will apply equation 1. That is limited by  $(8 d / t)$ .

And for that the argument is just by homogeneity and the fact that when you apply it once that equation in every summand somewhere there is a degree  $(t / 8)$ . In fact twice in every summand. So if, so even after number of many such iterations, in the end what do you see? You see that  $f$  is equal to sum of products and if in every product you have if there were let us say 1 iterations then there are 1 times  $(t / 8)$  contributions in each of these summands.

And this  $t / 8$  cannot exceed  $d$ . So that should give you the  $(8d / t)$  upper bound. But once the degree falls below  $t$  then you would not apply equation 1 because then you are then you cannot guarantee anything about the factors. So you do this only for high

degree up to high degree. That is our bound  $t$ . So that is one part. That is actually the biggest chunk.

Other small thing is that this in the end the factors are degree  $t$  and this you will represent fully in terms of sum of monomials in a completely stupid way. So that will further take size  $s^t$ . So finally, the degree  $t$  degree less than equal to  $t$  factors can have at most  $n^t$  or maybe big  $O$  many monomials. So combining the two things this number of summands which is  $s^{d/t}$  and within each factor you are expanding  $n^t$  many monomials.

So that gives you the product as the size bound. So  $C$  converts to  $\Sigma\Pi\Sigma\Pi^t$  circuit with the top fanin  $s^{O(d/t)}$  and size  $s^{(t+d/t)}$ . Is that clear? So that is the result for depth 4.

(Refer Slide Time: 1:07:56)

Corollary: An  $n$ -var.  $d$ -deg poly  $f$  requires homog.  $\Sigma\Pi^{O(d/t)}\Sigma\Pi^t$  circuits of top-fanin  $n^{O(d/t)}$   $\Rightarrow$   $f$  requires arithmetic circuits of size  $n^{O(t)}$ .  
(Super-poly. l.s. approach)

So what this means in terms of lower bound is just look at the contrapositive of what we had shown. So what we had shown is that if there is a small circuit then there is a small I mean there is a small “depth 4 circuit” also. So the contrapositive would say that if for an  $d$  degree polynomial  $f$  requires homogeneous  $\Sigma\Pi^{O(d/t)}\Sigma\Pi^t$  circuits of size of top fanin in fact, not even size.

So if you can come up with a polynomial  $f$  that is homogenous and it requires in terms of the degree it requires  $\Sigma\Pi^{O(d/t)}\Sigma\Pi^t$  circuits with top fanin slightly more than that

bound  $n^{\omega(d/t)}$ . So  $n$  is the number of variables that also I need. So an  $n$  variate  $d$  degree polynomial  $f$  that requires this special representation of top fanin  $n$  to the more than  $d/t$  asymptotically more than  $d/t$  and strictly more than  $d/t$ .

Then what does it mean?  $f$  requires arithmetic circuits of size how much? What is the lower bound that you get? Just look at the contrapositive. So if there is a circuit of size  $n^{O(d/t)}$ . So then you would have done in  $\text{poly}(n)$ . So now you get  $n^{\omega(1)}$ . You get super polynomial bound here. Maybe I should add a  $t$  as well somewhere. Should I add it here.

No. So this plus  $t$  was for the size sorry, the previous thing is correct. Just this, just if you prove something for the top fanin then you get a result. So what we have shown previously is actually stronger. It is saying that the top fanin in the representation is only it is  $s^{d/t}$ . So in this language it is if you prove a lower bound of  $n^{\omega(d/t)}$ , then that would mean that the arithmetic circuits in general for your  $f$  are super poly in the number of variables.

So that is a super polynomial lower bound approach. It is still not an exponential lower bound approach. It will not show that the polynomial is as hard as  $2^n$  which could be your final goal. But even super polynomial lower bounds are completely out of reach currently.

This is one way to highly specialize that question. You specialize to just very moderately high, moderately high top fanin depth 4 circuits and very well structured. So there is homogeneity and everything here.