– OPEN: Can we prove a similar homogenization for formulas?

Partial derivatives

– Let ∂_{x_i} denote the partial derivative operator (wrt x_i, $i \in [n]$).

We know that $\partial_{x_i} : \mathbb{F}[\bar{x}] \to \mathbb{F}[\bar{x}]$ is an $\mathbb{F}$-linear operator & has a product (Leibniz) rule :

$$\partial_{x_i}(fg) = f \cdot \partial_{x_i} g + g \cdot \partial_{x_i} f .$$

Theorem [Baur, Strassen '83]: Let $C(\bar{x})$ be a size-s, depth-d circuit. Then, there is a circuit $D(\bar{x})$, size-$O(s)$, depth-$O(d)$, that simultaneously computes $\partial_{x_i} C$, $i \in [n]$.

Proof:

• We prove the existence of D by induction on s.

- If C is a variable then we are done.
- Else let v be the deepest gate in C & denote its children by u, w.

Assume that u, w do not feed to gates other than v.

- Consider the circuit $C_{v=y}$ where the gate (subtree) v is replaced by a new variable y.

$C_{v=y}$ is smaller in size than C.

$\Rightarrow \exists$ a circuit D' of size $\alpha(s-1)$ [α is some constant] computing

$$\partial_{x_1} C_{v=y}, \ldots, \partial_{x_n} C_{v=y}, \partial_y C_{v=y}.$$

- Let $f, f_v, f_{v=y}$ be the output of $C, v, C_{v=y}$ resp.

Let X' be the variables that appear in the circuits for u & w. (Note: $|X'| \leq 2$.)

- Note that $f = f_{v=y}|_{y=f_v}$.

So, write $f = \sum_i a_i y^i = f_{v=y}(\bar{x}, y=f_v)$.

$$\Rightarrow \partial_{x_j} f = \sum (\partial_{x_j} a_i \cdot y^i + a_i \cdot \partial_{x_j} y^i)|_{y=f_v}$$
$$= (\partial_{x_j} f_{v=y})|_{y=f_v} + (\partial_y f_{v=y})_{y=f_v} \cdot \partial_{x_j} f_v.$$

- Therefore, for $x_i \notin X'$, $\partial_{x_i} f = (\partial_{x_i} f_{v=y})_{y=f_v}$.
- Since, X' has at most two variables, we can compute $\{\partial_{x_i} f \mid x_i \in X'\}$ by adding a constant ($\leq \alpha$) many gates & using D'.

$$\Rightarrow \text{size}(D) \leq \alpha \cdot (s-1) + \alpha = \alpha \cdot s.$$

- The depth(D) gets bounded by the induction argument as well. (Exercise.) $\square$

- This theorem suggests that a circuit C computes (almost) $\partial_{x_i} f$ while computing f.

  We will use this theme extensively to achieve "depth reduction".

- OPEN: Could all the second-order derivatives be computed in $O(s)$ size?

- This question is related to fast matrix multiplication:

  Consider the polynomial

$$C(\bar{x},\bar{y},\bar{z}) = \bar{y}\cdot AB\cdot \bar{z}^T,$$
where $A = (x_{1,i,j})_{i,j\in[n]}$ &
$B = (x_{2,i,j})_{i,j\in[n]}$.

- Note that $size(C) = O(n^2)$.
- The 2nd-order derivatives of C wrt $\bar{y},\bar{z}$ are $\{\partial_{y_i}\partial_{z_j} C = (AB)_{ij} \mid i,j\in[n]\}$.

$\Rightarrow$ If they have a common size $O(n^2)$ circuit, then we have an optimal way to multiply matrices!

<u>Depth reduction for formulas</u>

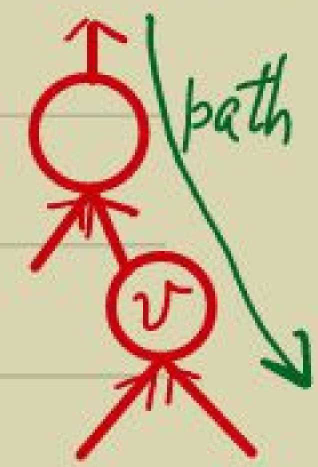
- Really interesting depth reduction theorems (& algos) are known. We warmup with formulas.

<u>Theorem</u> [Brent '74]: Let C be a size-s formula. There is an equivalent size-poly(s), <u>depth-$O(\lg s)$</u> formula. (unbounded fanin, fanout=1)

Proof: • Wlog assume fanin$(C) = 2$.

• Walk down from the root by taking the child whose subtree is larger.

Consider the <u>first</u> node v in this path whose formula size $\leq 2s/3$. Call this formula C_v.



$$\Rightarrow \frac{1}{2}\cdot\frac{2s}{3} \leq |C_v| \leq 2s/3 .$$

• Consider $C_{v=y}$ (i.e. formula v is replaced by a new variable y).

$$\Rightarrow \quad C = A\cdot C_v + B \quad \&$$

$$C_{v=y} =: \underline{A}\cdot y + \underline{B}, \text{ for polys } A, B \ \underline{\text{free}} \text{ of } y.$$

$$\Rightarrow \quad B = C_{v=y}|_{y=0}\ , \quad A = C_{v=y}|_{y=1} - B.$$

$$\Rightarrow C = \left(C_{v=y}(1) - C_{v=y}(0)\right)\cdot C_v + C_{v=y}(0). \quad \text{---} (1)$$

• Note that $|C_{v=y}| \leq s - |C_v| \leq 2s/3$

$\Rightarrow$ Eqn. (1) involves 4 formulas of size $\leq \frac{2s}{3}$.

- Thus, we get recurrences for the size & depth functions:

$$size(s) \leq 4 \cdot size(2s/3) + O(1),$$
$$depth(s) \leq depth(2s/3) + O(1).$$

$\Rightarrow$ $size(s) = poly(s)$ &
$depth(s) = O(\lg s)$.

$\square$

– In a general circuit there will be more overlap between $C_{v=y}$, C_v & so the above argument does not work.

– However, a different argument will work – based on recursively reducing the <u>degree</u> as we walk down.

<u>Theorem</u> (Valiant, Skyum, Berkowitz, Rackoff '83]: Let deg=d polynomial f be computed by a size-s circuit C. Then, there is a

poly(sd)-size, depth-$O(\log d)$ circuit C' computing f.

[Moreover, given C there is a poly(nsd)-time algorithm to construct C'.]

Proof: