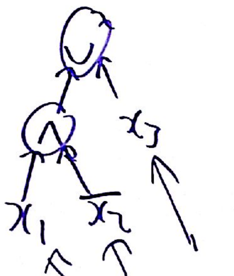
Formula over basis $\mathcal{B}$ w/ n vars:

- x_i + $\overline{x_i}$ are formulas
- F_1 & F_2 are formulas $\Rightarrow (F_1 \circledast F_2)$ is a formula.
  $\circledast \in \mathcal{B}$

Size: Total # of occurrences of $\circledast$'s, x_i's, and $\overline{x}_i$'s.

$(x_1 \wedge \overline{x_2}) \vee x_3 \rightarrow$ tree



leaves $= \#$ of x_i's $+ \overline{x}_i$'s

$L_{\mathcal{B}}(f) =$ min # of leaves in a $\mathcal{B}$-formula computing f.

Note size + leaves of a formula are within a factor of 2.

$$\ell \text{ leaves} \Leftrightarrow \text{size} \le 2\ell - 1$$

full binary tree
ℓ leaves $\Leftrightarrow$ size $2\ell-1$

Ind: ✓

Spse binary tree w/ size $2\ell-1$ / ℓ leaves. Add a node

leaves $= \ell+1$
size $= 2\ell-1+2$
$= 2(\ell+1)-1$

Thm (Riordan-Shannon) $\exists f \in \mathcal{B}_n$ s.t. $L_{\mathcal{B}_2}(f) \geq \dfrac{2^n}{6+\log n}$.

Pf. Counting. $L(n,\ell) = \# f \in \mathcal{B}_n$ w/ ℓ leaves

$$\le \left(\begin{array}{c}\#\text{ of full binary trees}\\ \text{w/ } \ell \text{ leaves}\end{array}\right) \cdot 16^{\ell-1} \cdot n^{\ell}$$

Generated by CamScanner from intsig.com

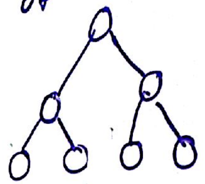
(full = every node has 2 or 0 children)

To specify a formula of ℓ leaves:
- "shape" (full binary tree w/ ℓ leaves)
- inner gates (16 choices for each leaf)
- inputs at leaf

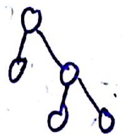
<u>Claim</u>: # of "shapes" is $\leq 4^\ell$.

Corresponds to # of strings w/ $\underline{\ell-1}$ pairs of well-balanced parentheses.

$$\frac{1}{\ell}\binom{2(\ell-1)}{\ell-1} \leq 2^{2(\ell-1)} < 4^\ell.$$



()()



(())

$$\text{So } L(n,\ell) \leq \underset{\uparrow \\ \#\text{trees}}{4^\ell} \cdot \underset{\uparrow \\ \text{inner gates}}{16^{\ell-1}} \cdot \underset{\uparrow \\ \text{inputs}}{n^\ell} \leq (64n)^\ell$$

$\exists f \in B_n$ s.t. $L_{B_2}(f) > \ell$, when $2^{2^n} > (64n)^\ell$

$\Leftrightarrow 2^n > \ell \log(64n)$

$\Leftrightarrow \dfrac{2^n}{\underset{(\log n + 6)}{\underset{\shortparallel}{\log(64n)}}} > \ell$ $\square$

Generated by CamScanner from intsig.com

$$\Pr_{f \in B_n}\left[L_{B_2}(f) \geq \frac{(1-\varepsilon)2^n}{\log n} \right] \geq 1 - o(1), \ \forall \varepsilon > 0.$$

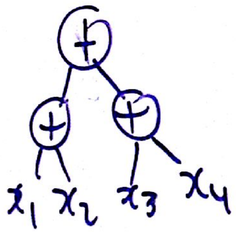
Thm [Lupanov] $L_{B_2}(f) \leq O(2^n/n), \ \forall f \in B_n$

Diff. between formulas & ckts: Basis matters more.

Know: $C_{B_1}(f) = \Theta(C_{B_2}(f))$, if B_1 & B_2 are complete.

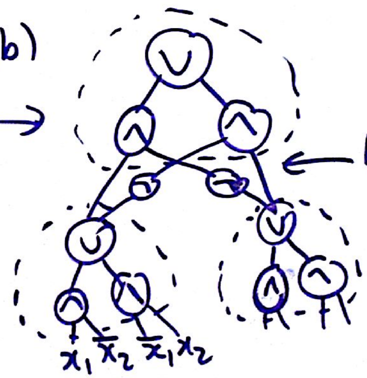
Leafsize can differ by poly factors.

Example: XOR_n. $C_{U_2}(XOR_n) = 3(n-1)$, $C_{B_2}(XOR_n) = n-1$.



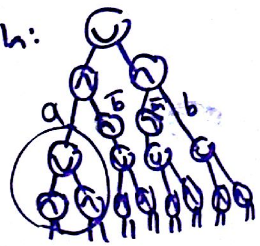
$$a \oplus b = (a \wedge \bar{b}) \vee (\bar{a} \wedge b)$$

$\longmapsto$



NOT a formula!

Formula:



For XOR_{2^d}, get a 4^d leaf formula.

$L(2^d) = 4 \cdot L(2^{d-1})$

$L(2) = 1$

But this isn't a pf!

Have to show there's no smaller formulas...

Generated by CamScanner from intsig.com

Polysize Formulas ≡ Log-depth Ckts

Know: Log-depth ckts can be turned into polysize formulas: $C(f) \le 2^{D(f)}$ proves $L(f) \le 2^{D(f)}$

Thm: $\exists c \geq 1$ s.t. $\forall$ formulas F (U_2/B_2) of size s, $\exists$ formula $G \equiv F$ of size $\le c \cdot s^3$, depth $\le c \cdot \log s$

i.e., $D(f) \le c \cdot \log L(f)$

Idea: Given F, "rebalance" the tree while preserving the fn F computes.

Find a node separator:

Lemma Let T be a full binary tree w/ ℓ leaves. $\exists$ subtree T' w/ leaves ℓ' s.t. $\lfloor \ell/3 \rfloor \le \ell' \le \lceil 2\ell/3 \rceil$.

Pf: Give an alg.

Define weight of node v = # leaves in subtree rooted at v

If $\ell = 2$, output a leaf as T'.

$v :=$ root of T

while $wt(v) > \lceil 2\ell/3 \rceil$,

{ set $v :=$ child of v w/ max wt. }

Output v.

Generated by CamScanner from intsig.com

<u>Alg halts</u>: Wt decreases as you move down the tree (wt of a leaf = 1)

Spse v is output. $\Rightarrow wt(v) \leq \lceil 2\ell/3 \rceil$

u = parent of v $\Rightarrow wt(u) > \lceil 2\ell/3 \rceil$.

x = sibling of v (has wt > 0) $\Rightarrow wt(v) \geq wt(x)$.

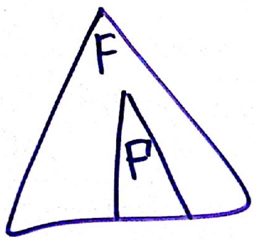
$$wt(v) + wt(x) = w(u) > \lceil 2\ell/3 \rceil$$

$$\Rightarrow 2\,wt(v) \geq wt(x) + wt(v) > \lceil 2\ell/3 \rceil$$

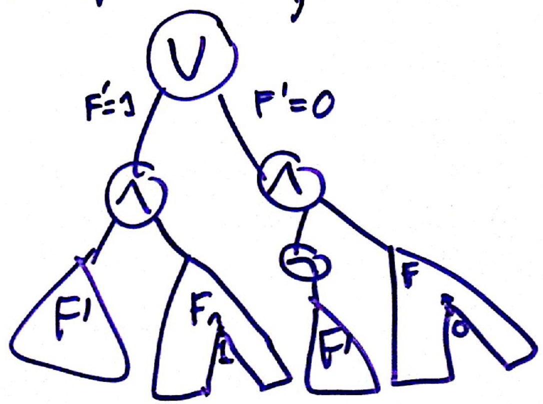
$$\Rightarrow wt(v) \geq \lfloor \ell/3 \rfloor. \quad \square$$

---

1. Given F, find a subformula F' using the lemma.



2. <u>Guess</u> the output of F'; check the guess.



3. <u>Recurse</u> on the four subformulas.

$$d(F) \le 2 + \max\{d(F'), d(F|_{F'=1}), d(F|_{F'=0})\}$$

$D(\ell)$ = depth of a formula w/ ℓ leaves

$$D(\ell) \le 2 + D(\lceil 2\ell/3 \rceil)$$

$$\Rightarrow D(\ell) \le 2 \cdot \log_{3/2} \ell + O(1)$$

$$\le 4 \cdot \log_2 \ell + O(1).$$

Size of new formula: $S(\ell) \le 4 \cdot S(\lceil 2\ell/3 \rceil) + O(1)$

$$S(\ell) \le O(\ell^{\frac{\log 4}{\log 3/2}})$$

≈ 3.42

Cor: For any $O(1)$-fanin complete bases B_1, B_2,

$$L_{B_1}(f) \le \text{poly}(L_{B_2}(f)).$$

Pf: $D_{B_1}(f) = O(D_{B_2}(f))$ (1)

$\exists c, d$ $\quad d \log(L_{B_i}(f)) \underset{(2)}{\le} D_{B_i}(f) \underset{(3)}{\le} c \log(L_{B_i}(f))$

(2) since $D(f) \ge \log L(f)$

(3) above thm

$$L_{B_1}(f) \le 2^{O(D_{B_1}(f))}$$

$$\underset{\text{by (1)}}{\overset{\text{by (2)}}{\le}} 2^{O(D_{B_2}(f))}$$

$$\underset{\text{by (3)}}{\le} \text{poly}(L_{B_2}(f)).$$

Generated by CamScanner from intsig.com

Q: Given formula of size s,

$\exists$ equiv. formula of $s^{1+\varepsilon}$ size

$(1+\varepsilon)\log_2 s$ depth, $\forall \varepsilon > 0$?

<u>Open</u> !?

Best I can find: [Bshouty-Cleve-Eberly, Bonet-Buss '91]

$\forall$ formulas over U_2 w/ ℓ leaves

$\exists$ equiv. formula w/ $\ell^{1+\varepsilon}$ leaves

$\leq \underbrace{(3 \ln 2)}_{\approx 2.08} \log_2 \ell$ depth, $\forall \varepsilon > 0$.

What can be computed w/ formulas / log-depth ckts?

<u>Addition</u> (n-bits): $O(n)$ size, $O(\log n)$ depth. ($\exists$ depth-3 $2^{O(\frac{n}{\log n})}$ size...)

<u>Mult</u>: $O(n \log n \log\log n)$ size, $O(\log n)$ depth

(Use depth reduction to show $O(n)$ size is impossible??)

<u>Division</u>: Given x, y n-bit integers, compute n MSBs of $\lceil x/y \rceil$.

[Beame-Cook-Hoover '80s] Division $\in NC^1$

Generated by CamScanner from intsig.com

FormulaEval $= \{(F, a) \mid$ formula F on input a outputs $1\}$.

<u>Thm</u> [Buss '89] FormulaEval $\in NC^1$

Can evaluate formulas with <u>larger formulas</u>
which can be eff. generated

Limits on formula size. Start with B_2.

<u>Def.</u> $SA_{2^k}(x_1, \dots, x_{2^k}, a_1, \dots a_k) = x_{\underbrace{bin(a_1 \dots a_k)}_{i}}$
"storage access"

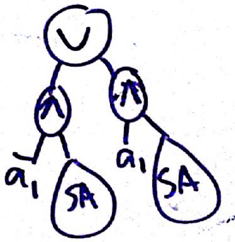
$C_{B_2}(SA_{2^k}) = 2 \cdot 2^k \pm O(2^{k/2})$

<u>Prop.</u> $C_2(SA_{2^k}) \leq 3 \cdot 2^k$

<u>Pf:</u> $SA_{2^k}(x_1, \dots, x_{2^k}, a_1, \dots, a_k) = (\overline{a_1} \wedge SA_{2^{k-1}}(x_1, \dots x_{2^{k-1}}, a_2, \dots, a_k))$
$\vee (a_1 \wedge SA_{2^{k-1}}(x_{2^{k-1}+1}, \dots, x_{2^k}, a_2, \dots, a_k))$.

$SA_1(x_1) = x_1$

$size(2^k) \leq 2 \cdot size(2^{k-1}) + 3$
$\leq 3 \cdot 2^k$



Generated by CamScanner from intsig.com