Lecture 16

Last class:

Concrete Complexity:

Circuits: NC^1 NC^2 (uniform-~~B~~$NC^1 \subseteq L \subseteq NL \subseteq NC^2$)

~~Branchi~~ Branching programs (non-uniform logspace)

Formulae (tree-like circuits)

$C(f) \leq O(BP(f)) \leq O(L(f))$

$NC^1 \equiv$ poly-sized formulae

Today:

Barrington's Theorem:

A languge is in $NC^1 \Leftrightarrow \exists\, O(1)$-width polysize BP compute it

Neciporuk's Thm:

Explicit function f in P s.t. $BP(f) \geq \Omega\left(\frac{n^2}{\log^2 n}\right)$ ($f: \{0,1\}^n \to \{0,1\}$)

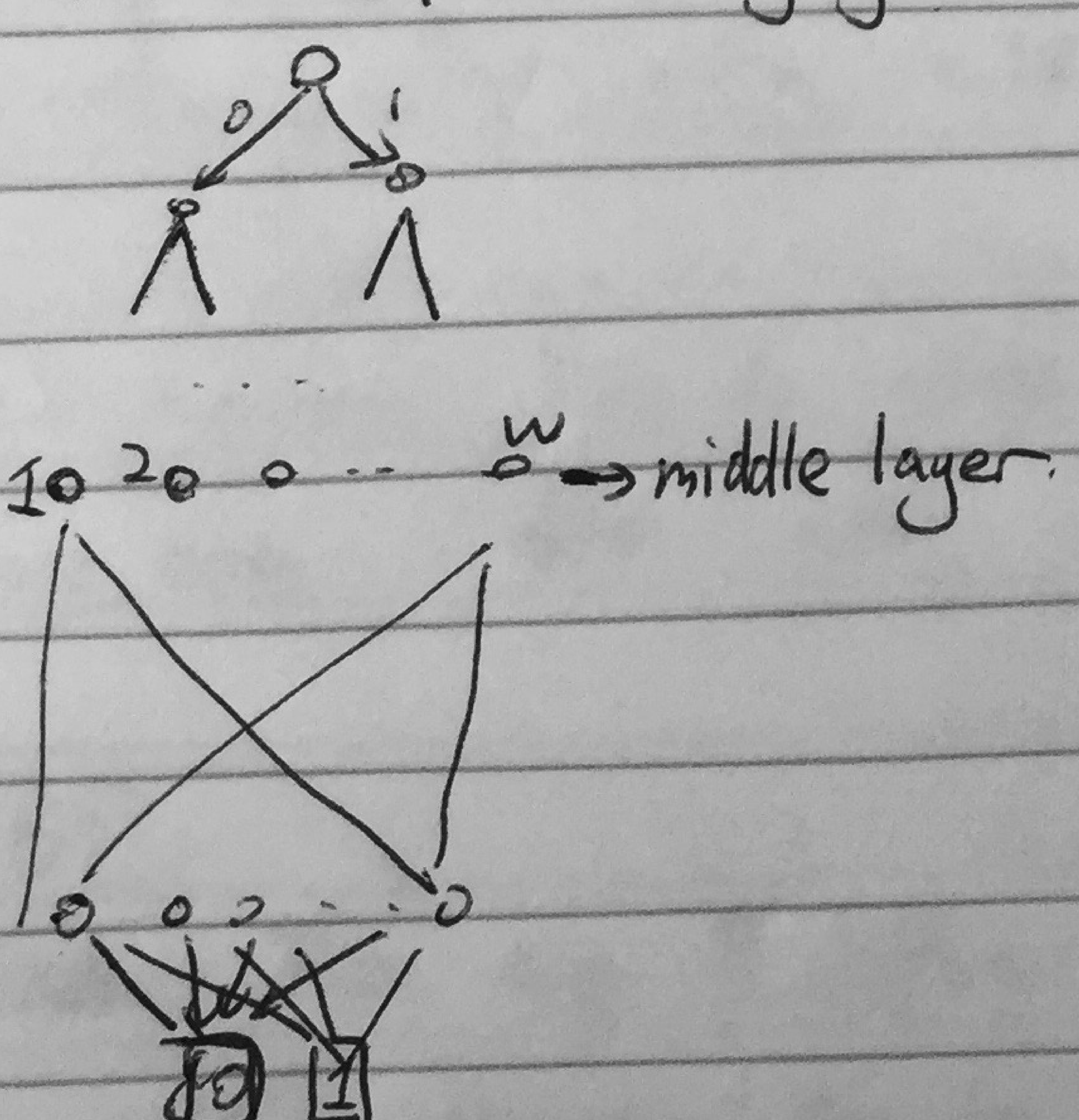
Comment:

We also know explicit function $g: \{0,1\}^n \to \{0,1\}$ in P s.t.

$$L(g) \geq \tilde{\Omega}(n^3)$$

Pf of Barrington's Thm:

~~For~~ a (constant) width-w BP $\Rightarrow$ language in NC^1



We can write it as

$$\bigvee_{i=1}^{w} (T_i \wedge B_i)$$

T_i: you reach node i

B_i: from node i you proceed to accept

By induction, ~~of~~ you will get a log depth circuit with fan-in w.

$NC^1 \Rightarrow O(1)$-width BP:

Barrington's proof:

- permutation BPs
- S_5 is not solvable
- Width 5 BP

Here we'll show:

Width-8 with simpler proof via SLP

Def: Straight line program on a register machine (SLP)

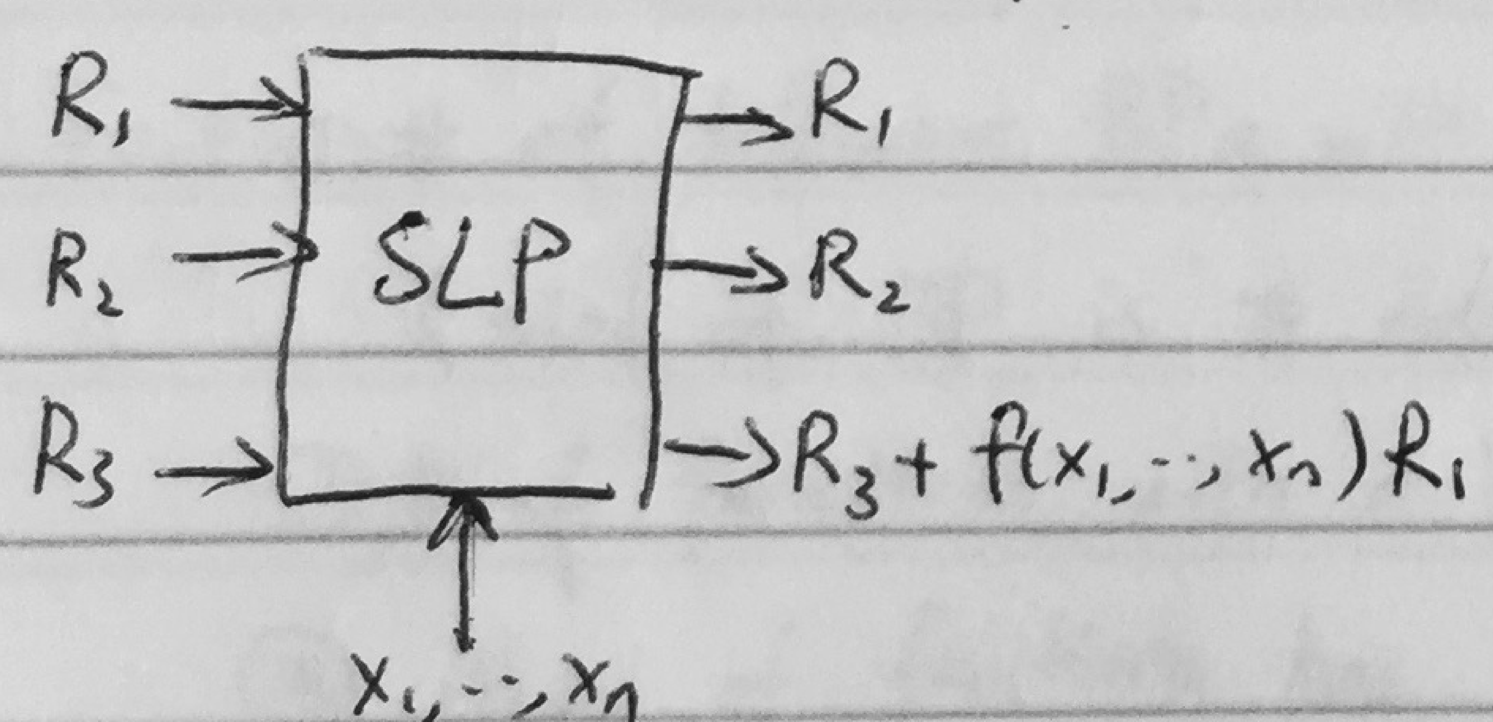
Input: $x_1, x_2, \ldots, x_n$ Register: $R_1, R_2, \ldots, R_m$

Operation: $R_j \leftarrow R_j + R_i$ (Here "+" is XOR function)

$R_j + x_\ell R_i$

$R_j + \bar{x}_\ell R_i$

$f(x_1, \ldots, x_n) \in NC^1$, depth d formulae, SLP with 3 registers to "compute" f.



Induction hypothesis: Can do above with SLP within 6^d instructions

Basic case: $d=0$ easy

$SLP \Rightarrow BP$:

~~Each~~ + There are 8 different status for 3 registers

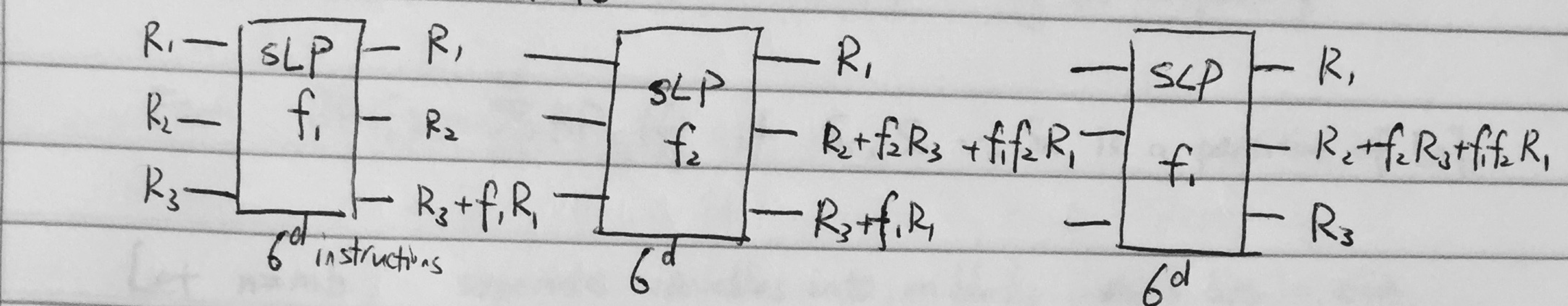
The instruction can be seen as transform to the next layer

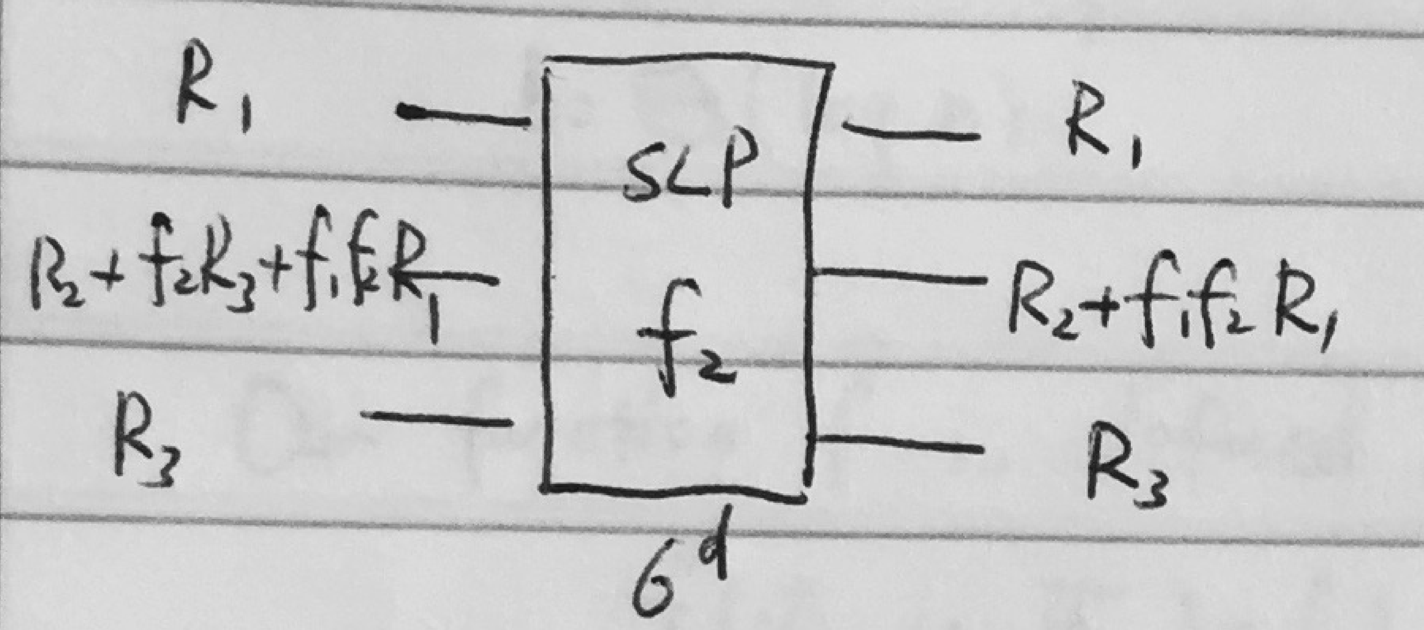
the query is x_ℓ in the operation

So we get a width-8 BP with 6^d layers

The induction in ($NC^1 \Rightarrow SLP$):

Case 1: $f = f_1 \wedge f_2 = f_1 \cdot f_2$





at most $4 \cdot 6^d$.

Case 2: $f = f_1 \vee f_2 = f_1 f_2 + f_1 + f_2$

at most $6 \cdot 6^d + 2 = 6^{d+1} + 2$

Neciporuk lower bound:

Fact: ① # of different BPs with size $S \leq S^{O(S)}$

(size of BP is # nodes in BP)

② # of functions with n bits is 2^{2^n}

③ Most of functions has $BP(f) \geq \Omega(\frac{2^n}{n})$

Idea: define $f_a(y) = f(y, a)$ where $f: \{0,1\}^n \to \{0,1\}$, $y \in \{0,1\}^b$, $a \in \{0,1\}^{n-b}$

$f_a: \{0,1\}^b \to \{0,1\}$

Choose b = ~~(logn)~~ $\leq \log n$: $\# f_a \leq 2^n$

There fore: ~~f has~~ any BP of f has a ~~at least~~ $\Omega(\frac{n}{\log n})$ nodes labelled $\{x_1, \ldots, x_b\}$

And we can get $\frac{n}{\log n}$ blocks

$BP(f)$: $f(x_1, \dots, x_n)$. $S \subseteq \{1, \dots, n\}$

define $BP_S(f)$ = min # nodes $\{x_i \mid x_i \in S\}$ in any BP computes f

Fact: $BP(f) \geq \sum_{i=1}^{m} BP_{S_i}(f)$ if $S_1, S_2, \dots, S_m$ is a partition of $[n]$

Let $n = mb$, seperate variables into m blocks with b bits in each.

$b = \Theta(\log n)$

Our function f is defined as: (Element Distinct (ED))

$$f(\bar{y}_1, \dots, \bar{y}_m) = \begin{cases} 1 & \text{if } \bar{y}_i \neq \bar{y}_j \ \forall i \neq j \\ 0 & \text{otherwise} \end{cases}$$

where $\bar{y}_1, \dots, \bar{y}_m \in \{0,1\}^b$.

Claim: For each ~~ehoice~~ choice of $\bar{y}_2, \dots, \bar{y}_m$ which are different values, the reduced function on $\bar{y}_1$ is distinct

So the number of different reduced functions on $\bar{y}_1$ is at least

$$\binom{2^b}{m-1} \geq \left(\frac{2^b}{m}\right)^{m-1} \geq 2^{b(m-1) - m\log m}$$

~~BP~~ By simple counting:

$BP_{S_1}(ED) = L$ then $L^{O(L)} \geq M = 2^{b(m-1) - m\log m}$

then $L \geq \Omega\left(\frac{\log M}{\log \log M}\right) \geq \Omega\left(\frac{b(m-1) - m\log m}{\log(b(m-1) - m\log m)}\right)$ $\cancel{\geq \Omega\left(\frac{n}{\log n}\right)}$

If we choose $b = 2\log n$, $m = \frac{n}{b}$

$b(m-1) - m\log m \geq \frac{n}{2}$.

Therefore

$BP_{S_1}(ED) \geq \Omega\left(\frac{n}{\log n}\right)$

Therefore $BP(ED) \geq \sum_{i=1}^{m} BP_{s_i}(ED) \geq \Omega(\frac{n}{\log n}) \cdot \frac{n}{2\log n} \geq \Omega(\frac{n^2}{\log^2 n})$

Lower bounds against bounded-depth circuits:

AC^0: functions that can be computed by $\underbrace{\text{constant depth circuits}}^{\text{poly size}}$ with AND/OR gates of unbounded fan-in. and NOT gates.

Example: $PARITY(x_1, \dots, x_n) = x_1 \oplus \dots \oplus x_n$

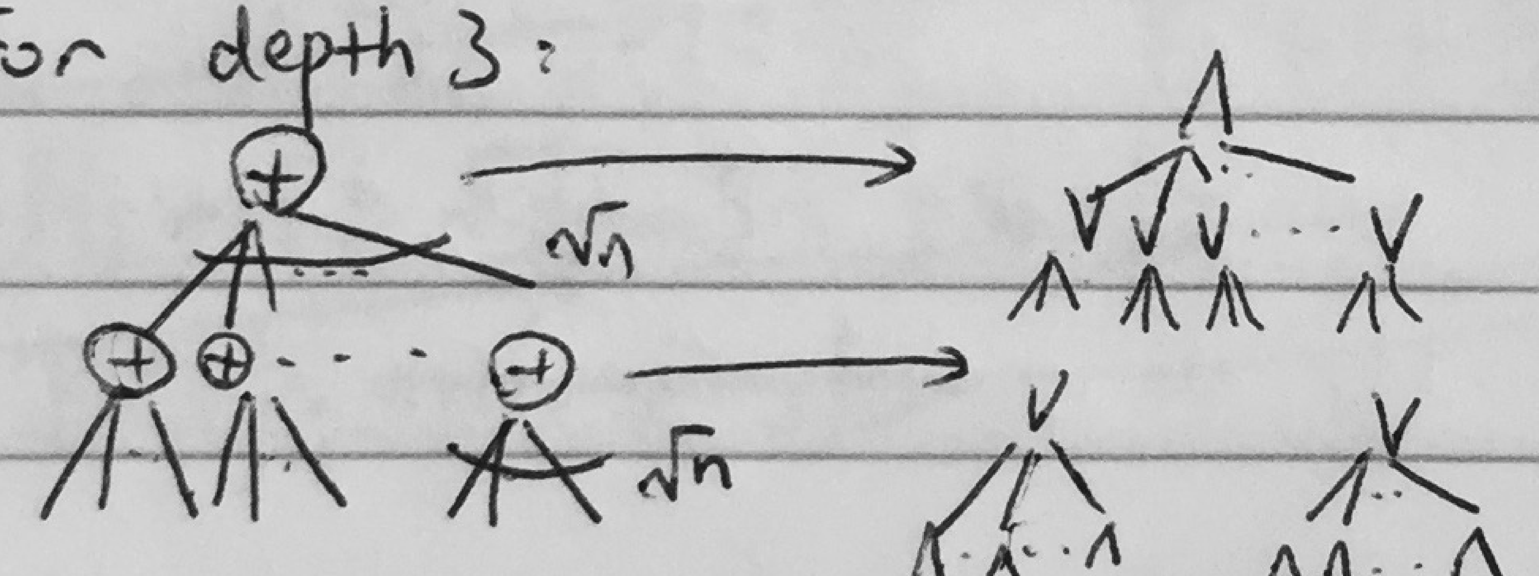
For depth-~~th~~2 circuit: need size $\tilde{\Omega}(2^n)$

Note: $x_1 \oplus x_2 = (x_1 \vee x_2) \wedge (\neg(x_1 \wedge x_2))$

$PARITY(x_1, \dots, x_n) = PARITY(x_1, \dots, x_{n/2}) \oplus PARITY(x_{n/2+1}, \dots, x_n)$

- Therefore $PARITY(x_1, \dots, x_n) \in NC^1$

For depth 3:



PARITY can be written as CNF/DNF of size 2^n

Therefore we get depth-3 circuit with size $2^{O(\sqrt{n})}$

Similarly we can get $2^{O(n^{\frac{1}{d-1}})}$-sized circuit with depth d can compute PARITY

[Furst - Saxe - Sipser, Yao, Håstad 86']

If a depth d AC^0 circuit compute PARITY on n variables, its size is $2^{\Omega(n^{\frac{1}{d-1}})}$.