Nechiporuk

Tuesday, March 1, 2022 17:33

What makes a function easy/hard?

e.g. PAR: if you look locally, two "roles":

- either parity of the rest is 0
  - $\Rightarrow$ PAR $= 1$ iff parity over S is 1
- or parity of the rest is 1
  - $\Rightarrow$ PAR $= 1$ iff parity over S is 0

One possible answer:

hard "if groups of variables play many roles"

subfunctions

$f : \{0,1\}^n \to \{0,1\}$

Let
- $I \subseteq [n]$ be a set of vars
- $w \in \{0,1\}^{[n] \setminus S}$ be an assignment to <u>other</u> vars.

Then

$$f_{I|w} : \{0,1\}^I \to \{0,1\}$$

$$f_{I|w} : \{0,1\}^I \to \{0,1\}$$

$$x \to f(x, w)$$

plug x into I, w into $[n] \setminus I$

"Number of roles" of I:

$$\#\mathrm{sub}(f, I) := \#\{f_{I|w} \mid w \in \{0,1\}^{[n] \setminus I}\}$$

Examples

$\forall I \neq \emptyset$ $\quad \#\mathrm{sub}(\mathrm{PAR}, S) = 2$

(as we saw)

$\forall I$ $\quad \#\mathrm{sub}(\mathrm{MAJ}, S) \leq |I|+1$

(b/c only the Hamming weight matters)

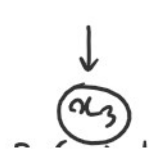
$\forall f, I$ $\quad \#\mathrm{sub}(f, S) \leq 2^{n-|I|}$

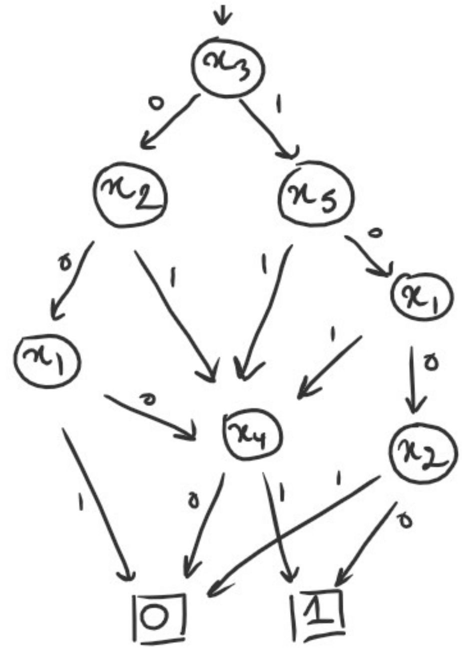
(b/c only that many possible strings y)

$$\#\mathrm{sub}(f, I) \leq 2^{2^{|I|}}$$

(b/c specified by value on each of the $2^{|S|}$ inputs)

Why does #sub matter?

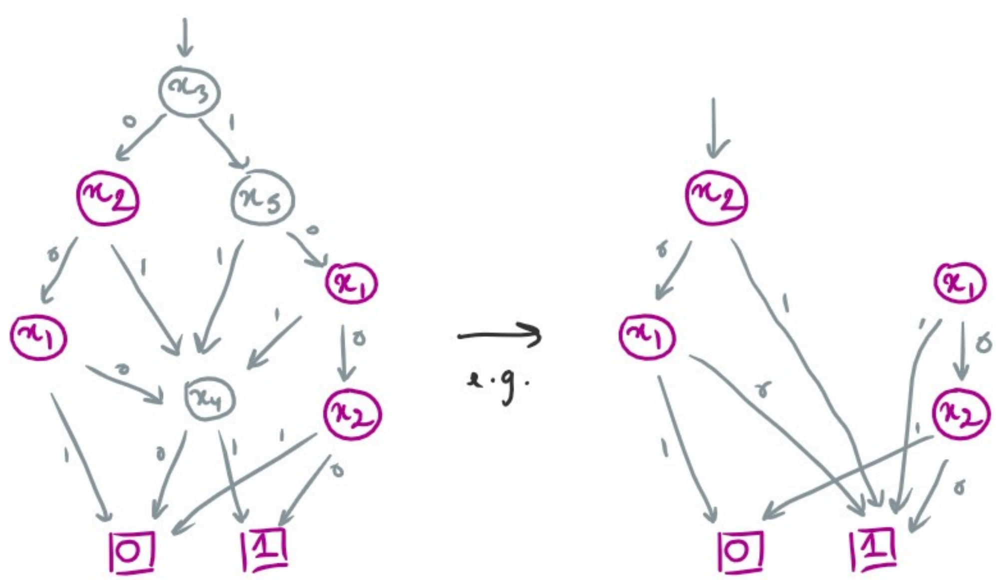




<u>Lemma</u> If vars of I queried s times, then

$$\#\text{sub}(f, I) \le s^{O(s)}.$$ (i.e. few possible values for $f_{I|y}$)

<u>Proof</u> Can "short-circuit" all nodes querying other vars



And there's only $s^{O(s)}$ possible BPs on s (given)

And there's only $s^{O(s)}$ possible BPs on s (given) nodes: just need to know where the $O(s)$ links go. ∎

Corollary $s \geq \dfrac{\log(\#sub(f, I))}{\log\log(\#sub(f, I))}$.

Getting a LB

Thm Let $I_1, \dots, I_\ell \subseteq [n]$ be disjoint.

Then a BP computing f has size

$$\geq \Omega\left(\sum_{r=1}^{\ell} \frac{\log(\#sub(f, I_r))}{\log\log(\#sub(f, I_r))} \right)$$

Proof Apply corollary for each I_r.

How to maximize this?

We know $\forall I$, $\#sub(f, I) \leq \min(2^n, 2^{2^{|I|}})$.

Assume the best: try to get $\#sub(f, I) = \min(2^n, 2^{2^{|I|}})$

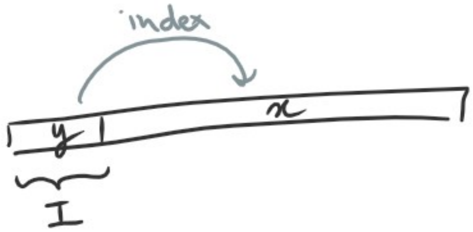
$\Rightarrow$ no point in making $|I| > \log n$.

Can get $\frac{n}{\log n}$ groups, each with $\dfrac{\log(\#sub)}{\log\log(\#sub)} = \dfrac{n}{\log n}$.

$\Rightarrow$ get $\Omega\left(\frac{n^2}{\log n}\right)$ at best. (by Jensen's)

Embedding one random subfunction

Embedding one random subfunction



$x \in \{0,1\}^n$

$y \in [n] = \{0,1\}^{\log n}$

$\text{INDEX}(x, y) = x_y$

Then $\#\text{sub}(f, I) = 2^n$. :)

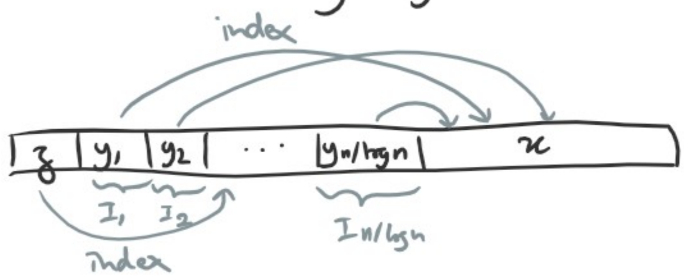
By fixing x to different values, can get

all possible subfunctions over I

$\approx$ the subfunction over I is a "random function"

Embedding many random subfunctions

Idea: have many y's each active some of the time



$x \in \{0,1\}^n$

$y_r \in [n] = \{0,1\}^{\log n}$

$z \in [n/\log n] = \{0,1\}^{O(\log n)}$

$\text{MULTIINDEX}(x, \{y_r\}, z) = x_{y_z}$

Then $\forall j \in [n/\log n]$, $\#\text{sub}(f, I) = 2^n$:)

$\Rightarrow$ this function requires BPs of size $\Omega\left(\frac{n^2}{\log^2 n}\right)$.

$\Rightarrow$ this function requires BPs of size $\Omega\left(\frac{n^2}{\log^2 n}\right)$.

(still best known LB on general BPs)

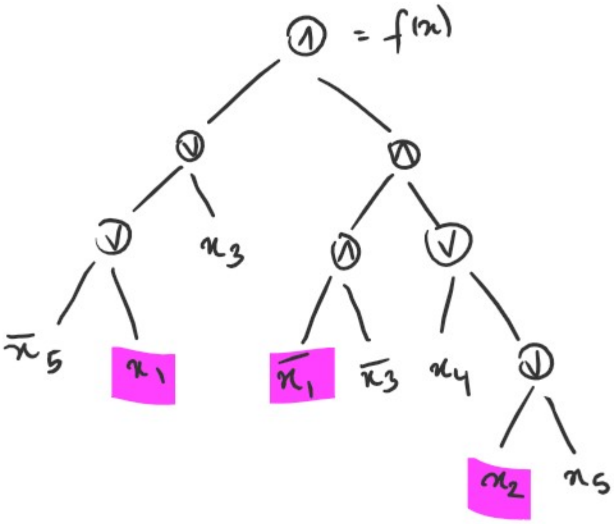
Shannon vs Nechiporuk

	Shannon	Nechiporuk
assumption	random function	all possible subfunctions
what do you LB?	size	# queries to I
technique	counting	counting
bound	$s^{O(s)}$	$s^{O(s)}$
input size	n	$\log n$
conclusion	$s \geq \frac{\log \#\text{fun}}{\log\log \#\text{fun}} = \frac{2^n}{n}$	$s \geq \frac{\log \#\text{sub}}{\log\log \#\text{sub}} = \frac{n}{\log n}$

$\left.\right\}$ for each I

Nechiporuk just uses the fact that "you can embed $\frac{n}{\log n}$ random functions of input size $\log n$ inside an explicit function"

Another model: formulas

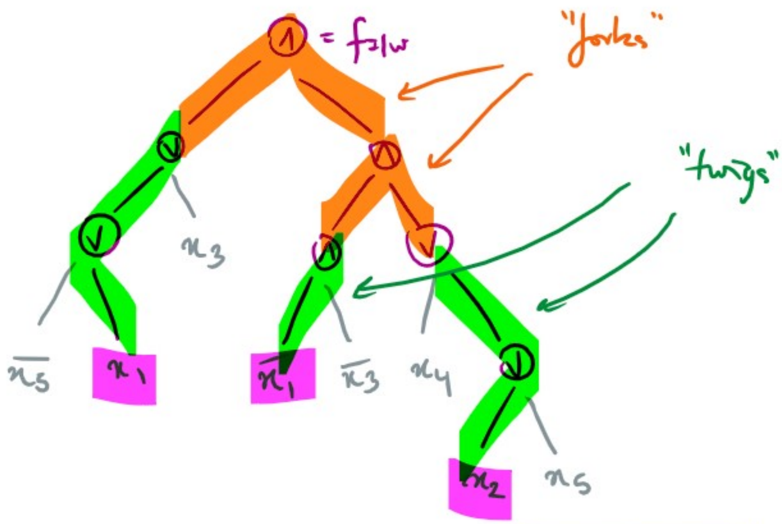


$I = \{1, 2\}$

Only $s = 3$ queries

<u>Lemma</u> If vars of I queried s times, then

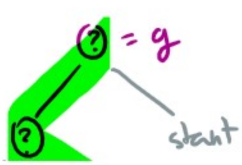
$$\#\mathrm{sub}(f, I) \leq 2^{O(s)}.$$



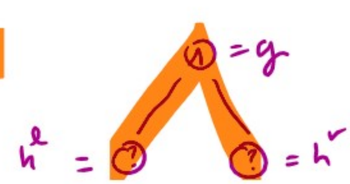
Idea: bound $\#\mathrm{sub}(f, I)$ by induction.

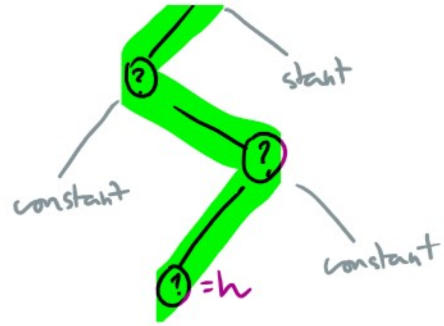
Two cases:

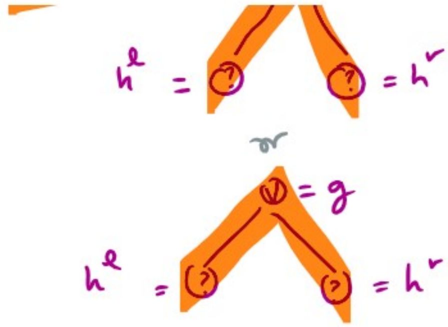
(a)



(b)







$$g_{I|w} \equiv \begin{cases} h_{I|w} \\ \neg h_{I|w} \\ 0 \\ 1 \end{cases}$$

$$\Rightarrow \#sub(g, I) \le 2 \cdot \#sub(h, I) + 2$$
$$\le 4 \cdot \#sub(h, I)$$

in either case, $g_{I|w}$ is fixed by $h^\ell_{I|w}$ and $h^r_{I|w}$

$$\Rightarrow \#sub(g, I) \le \#sub(h^\ell, I) \cdot \#sub(h^r, I)$$

And if s leaves, $\le O(s)$ twigs

$$\Rightarrow \le 4^{O(s)} = 2^{O(s)} \text{ possibilities.} \qquad \boxtimes$$

Corollary $s \geq \Omega(\log(\#sub(f, I)))$.

Getting a LB

Thm Let $I_1, \ldots, I_\ell \subseteq [n]$ be disjoint.

Then a formula computing f has size

$$\geq \Omega\left(\sum^{\ell} \log(\#sub(f, I_r))\right).$$

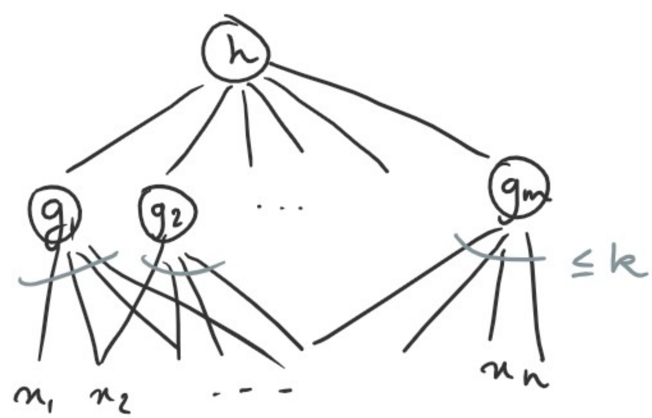
$$\geq \Omega\left(\sum_{r=1}^{\ell} \log(\#\text{sub}(f, T_r))\right).$$

Proof: Use corollary for each r.

$\Rightarrow$ MULTIINDEX needs formulas of size $\Omega\left(\frac{n^2}{\log n}\right)$.

Another model: composition complexity

Say you compute f as $h(g_1, \ldots, g_m)$.



How many inner functions do you need?

Defn: $CC_k(f)$ = min value of m

Trivial bounds: $CC_k(f) \geq \frac{n}{k}$ (must query all vars)

$CC_k(f) \leq n$ (could just "repeat the inputs")

Bounding #subfunctions of $f = h(g_1, \ldots, g_m)$

- $\#\mathrm{sub}(f, I) \le \prod_{j=1}^{m} \#\mathrm{sub}(g_j, I)$
- if g_j queries s vars of I, then

$$\#\mathrm{sub}(g_j, I) \le \min\left(2^{2^s}, 2^k\right)$$

$\Rightarrow$ by optimizing, get

$$CC_k(\text{MULTIINDEX}) \ge \Omega\left(\frac{n^2/\log n}{k^2/\log k}\right)^{*}.$$

* actually, min of that and $\Omega(n)$

But it says nothing when $k = \frac{n}{100}$ (say).

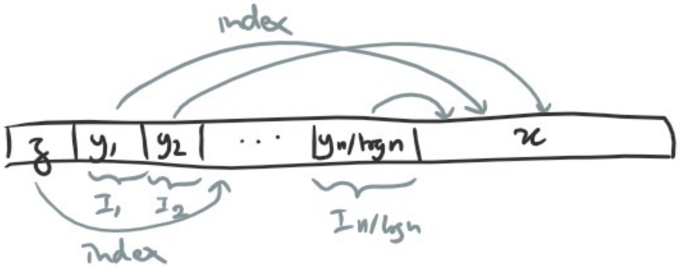
(and indeed, $CC_{\frac{n}{100}}(\text{MULTIINDEX}) = O(1)$)

Making MULTIINDEX harder

Issue : can make sure each I_r fits entirely within a single inner function g_j

$\Rightarrow$ huge number of subfunctions

Solution : Make "every" subset of variables have many subfunctions. The inner functions can't cover all of them!



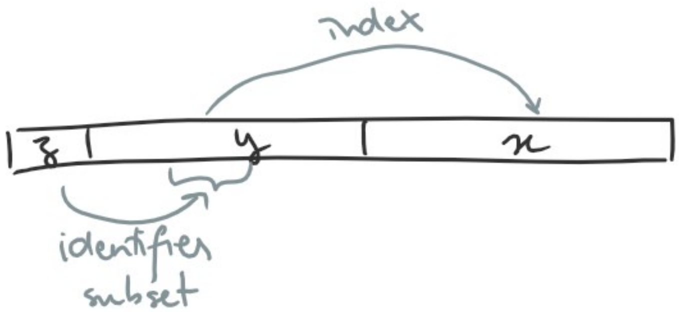
$x \in \{0,1\}^n$

$y_r \in [n] = \{0,1\}^{\log n}$

$z \in [n/\log n] = \{0,1\}^{\log(\frac{n}{\log n})}$

$$\textsc{MultiIndex}(x, \{y_r\}, z) = x_{y_z}$$

$$\downarrow$$



$x \in \{0,1\}^n$

$y \in \{0,1\}^n$

$z \in \binom{[n]}{\log n} = \{0,1\}^{O(\log^2 n)}$

$$\textsc{SubsetIndex}(x, y, z) = x_{\overline{\underbrace{y_z}}}$$

← y_z, viewed as an integer encoded in binary

↑ y at the subset of positions encoded by z

Consequence : $\forall I$, "subset of y" of size $\log n$, $\#\mathrm{sub}(f, I) = 2^n$.

Sketch of the rest

- choose a random I
- whp, each inner function queries only a

- whp, each inner function queries only a small fraction of I (say, $\frac{|I|}{10}$)
- so $\#\mathrm{sub}(f, I) \leq \prod \#\mathrm{sub}(g_j, I)$

$$\leq \left(2^{2^{|I|/10}}\right)^m$$

$$\Rightarrow m \geq \frac{n}{2^{|I|/10}} = n^{.9}$$

nice!