LECT 04

Note Title 2/19/2007

TODAY

1. NECIPORUK'S LOWER BOUND
2. BARRINGTON'S THEOREM
   (proof due to Ben-Or + Cleve)

Review

Last time: 4 Non-uniform Models of Computation

1. TMs with advice
2. Circuits
3. Branching Programs
4. Formulae

Resources

- # bits of advice ;
- advice TM. Time ;
- size
- depth
- width.

Counting bounds

if $\mathcal{F}$ is a family of functions

then $\exists\ f \in \mathcal{F}$ s.t.

$$\text{size}(f) \geq \Omega\left(\frac{\log(\mathcal{F})}{\log\log(\mathcal{F})}\right).$$

(Proof : # function of size $s \leq 2^{s \log s}$).

But : What is $NP \cap \{\{0,1\}^n \to \{0,1\}\}$?

Your answer here $\boxed{?}$

NECIPORUK'S THEOREM

$\exists f \in P$ s.t. BP-size $(f) \geq \left(\frac{n}{\log n}\right)^2$.

Proof Idea - 1 :

Will create function $f : \{0,1\}^k \times \{0,1\}^{n-k} \to \{0,1\}$

s.t. last $(n-k)$ bits really specify function on first k coordinates.

How: example

$f(i, \bar{x}) = x_i$ where

first i bits signify

index from $1 \ldots 2^k$

& $\bar{x}_1 \ldots x_{n-k} = 2^k$-bit string.

$\forall\ \bar{x} \neq \bar{y} \quad f_x(\cdot) \neq f_y(\cdot)$

where $f_x(i) = f(i, x)$.

Now BP for f gives BP for

f_x for every x.

$\#\ x's = 2^{n-k} \approx 2^n \Rightarrow$ BP.size$(f) \geq |\{f_x\}|$

$$\geq \frac{n}{\log n}$$

But this is sub-linear ... how to improve ?

<u>Idea 2</u> : for $S \subseteq [n]$

$\text{BP-size}_S(f)$ = # edges labelled with literals in S

Above proof actually implies

$$\underbrace{\text{BP-size}_{\{1..k\}}}(f) \geq \frac{n}{\log n}$$

$\uparrow$

<u>$\log n$ variables</u>

Can we repeat this for other blocks.

Well ... not for same f, but different one

FUNCTION: DISTINCT? $(\underbrace{X_1 \dots X_{1k}}_{U_1}, \underbrace{X_{21} \dots X_{2k}}_{U_2}, \dots \underbrace{X_{\ell 1} \dots X_{\ell k}}_{U_\ell})$, with $S_1, S_2, \dots, S_\ell$ written over the blocks

$$\text{DISTINCT?}(U_1, \dots U_\ell) = 1 \quad \text{if } \forall i \neq j \quad U_i \neq U_j$$
$$= 0 \quad \text{o.w.}$$

Claim: $\forall\, i$, # functions

$$\left| \left\{ f_{a_1 \dots a_\ell} (U_i) = \text{Distinct?}(a_1, \dots, U_i, \dots a_\ell) \right\} \right| \geq \binom{2^k}{\ell - 1}$$
$$\geq \left(\frac{2^k}{\ell} \right)^{\ell}$$

Claim: BP-size$_{S_i}$(Distinct) $\geq \dfrac{n}{\log n}$

Claim: BP-SIZE$(f) \geq \sum_i$ BP-size$_{S_i}(f)$

Putting Claims together: get

BP-size (Distinctness) $\geq \ell^2 k - \ell^2 \log \ell$

Setting $k = 2\log \ell$ & $n = \ell k$

get

$$\geq \Omega\left(\frac{n^2}{\log^2 n}\right)$$

$\boxtimes$ (NECIPORUK)

BARRINGTON (BEN-OR + CLEVE):

Motivation: • Can we use non-uniformity to prove $P \neq L$?

• Maybe we can argue that "simple" functions don't have small width BPs.

• But every CNF / DNF formula has width - 3 BP ... of exponential size...

• Really need to show that no poly-size BP exists for some function in P.

• Natural candidate: $\mathrm{MAJORITY}(x_1 \ldots x_n) = 1 \iff \sum x_i \geq \frac{n}{2}$.

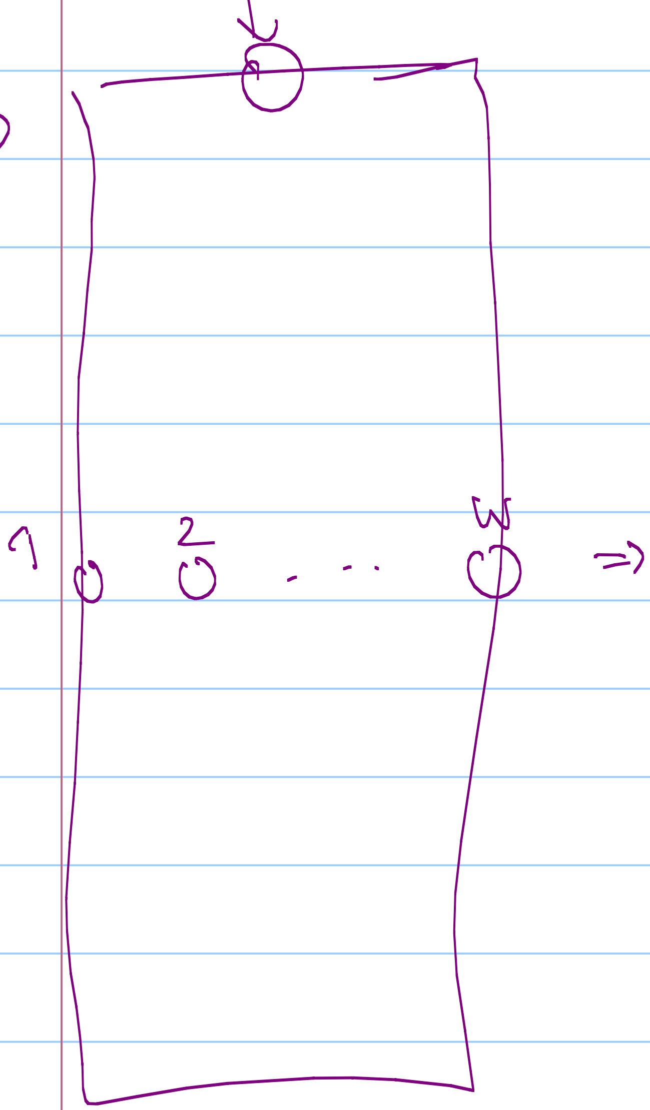
Barrington's Theorem:

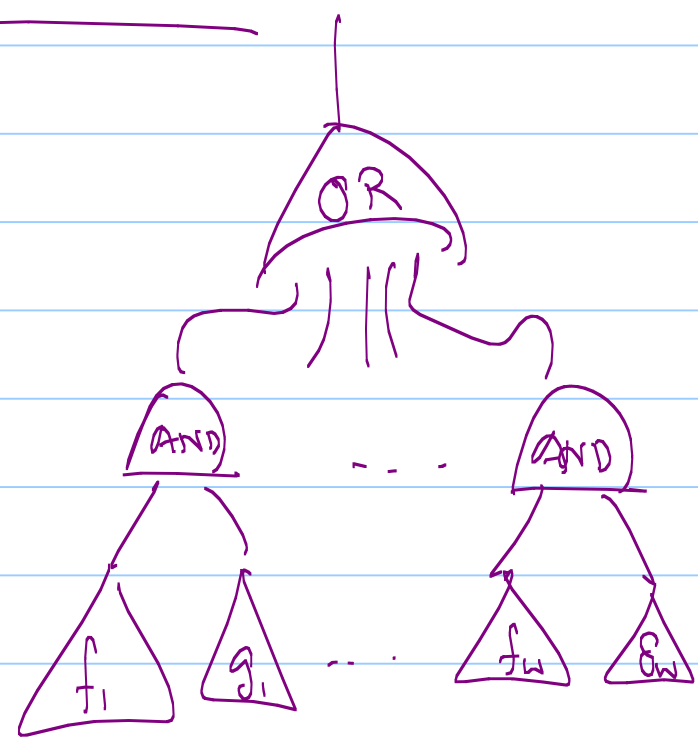
f has $O(1)$ width poly size BP

$\Longleftrightarrow$ f has log-depth formula (over any finite basis)

$\longrightarrow x$

$\Rightarrow$





(other bases similar).

In above

$f_i = 1$ if top b.p. reaches i^{th} state in middle level

$g_i = 1$ if bottom b.p. accepts starting at middle level.

$\Longleftarrow$ Much harder (unexpected)

Ben-Or + Cleve's proof : Simple idea.

"Strong Induction"

REGISTER Machines

Given by ℓ registers $R_1 \ldots R_\ell$

& S instructions

I_1

I_2

$\vdots$

I_S

I_j : of the form $R_i \leftarrow R_j + R_k \star R_\ell$

or $R_i \leftarrow R_j \pm X_k \star R_\ell$

Register Machine Computes $f(x_1 \ldots x_n)$

if it maps.

$$(R_0 \ldots R_\ell) \to (R_1 \ldots R_{\ell-1},\ R_\ell + f(\,) \cdot R_\ell)$$

<u>Strong Hypothesis</u>: if f has depth d

$\{$AND, NOT$\}$ formula then

$\exists$ size 4^d, 3 register machine

<u>Prop</u>: f has size S, ℓ-Register m/c

$\Rightarrow$ f has size $O(S)$, 2^ℓ width BP

(8 in our case)

Proof:

f computed by $M_1 = I_1 \; \vdots \; I_s$

$\Rightarrow (1-f)$ computed by

$I_1' \; \vdots \; I_s'$ $\quad I_j' = \overline{I_j}$ replace + by $-$.

$I_{s+1} = R_\ell \leftarrow R_\ell + R_1$

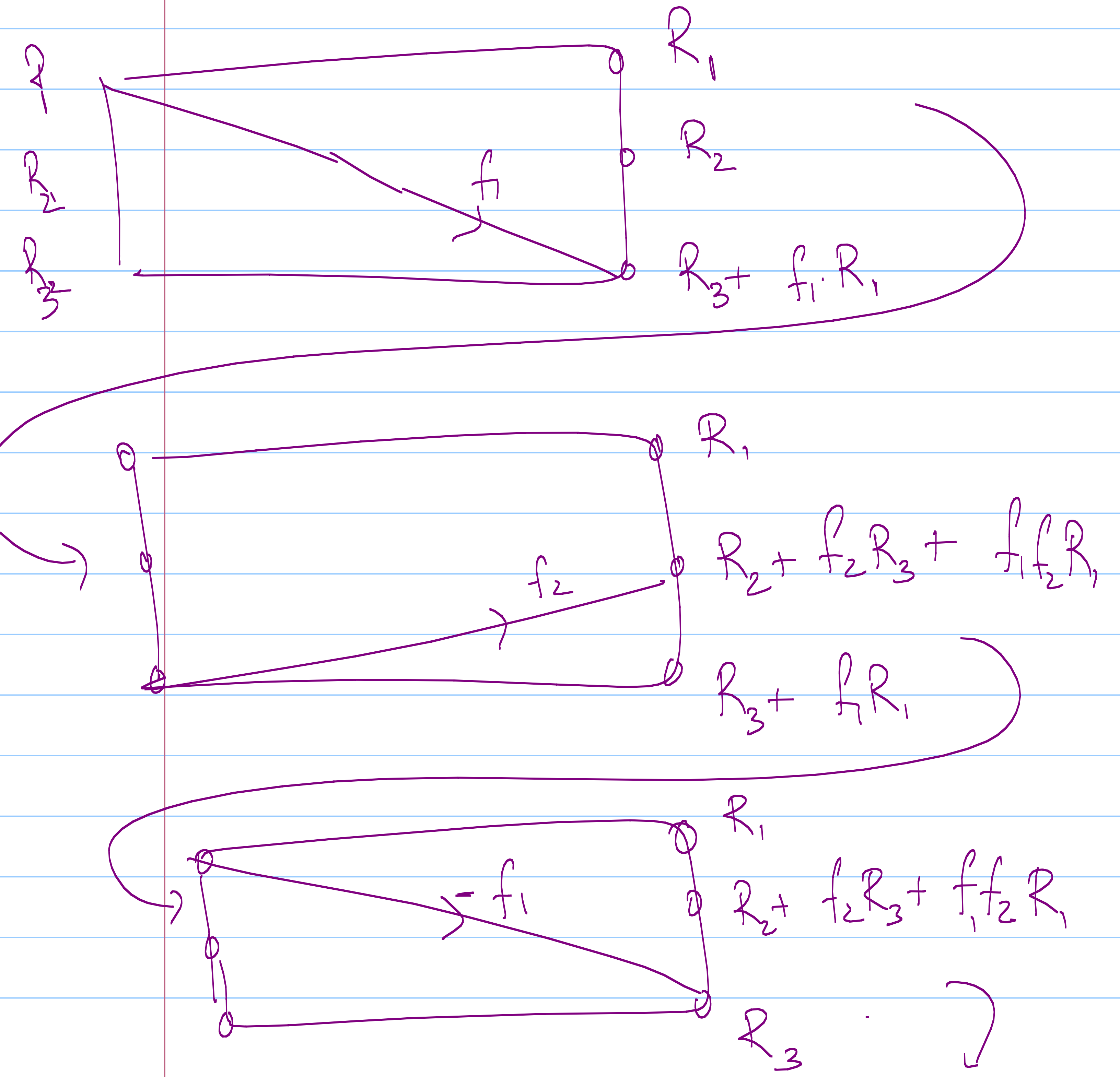
(Previously R_ℓ had $R_\ell^0 - fR_1^0$

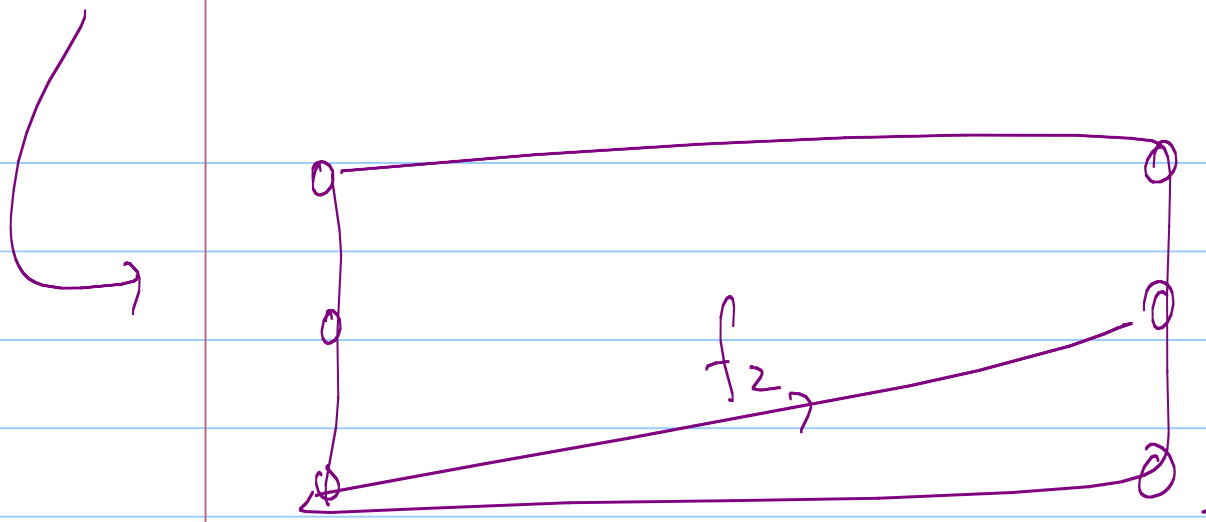
& R_1 " R_1^0

Now: $R_\ell \leftarrow R_\ell^0 + (1-f) R_1^0$)

Interesting Case

$$f = f_1 \wedge f_2$$





$R_2 + f_1 f_2 R_1$