Today

1. Ncicork's Lower Bound

2. Barrington's Theorem
   (proof due to Ben-Or + Cleve)

Review

Last time: A Non-uniform Models of Computation

1. Trees 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.
\[
\text{size}(f) \geq \Delta \left( \frac{\log(\mathcal{F})}{\log \log(\mathcal{F})} \right).
\]
(Proof: # function of size $s \leq 2^{s \log s}$)

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

Your answer here

NECPORUK'S THEOREM

\[ \exists f \in \mathbb{P} \quad \text{s.t.} \quad \text{BP-size}(f) \geq \left( \frac{n}{\log n} \right)^2. \]

Proof: \text{deg} = 1:

will create function $f : \{0,1\}^k \times \{0,1\}^{n-k} \rightarrow \{0,1\}^n$.

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

How: Example

\[ f(i, \overline{X}) = X_i \quad \text{where} \]

\[ \text{first } i \text{ bits signify index from } 1 \ldots 2^k \]

\[ \& \quad \overline{X}_1 \ldots \overline{X}_{n-k} = 2^{k-b} \text{ bits string}. \]

\[ \forall \overline{x} \neq \overline{y}, \quad f_x(\cdot) \neq f_y(\cdot) \]

where \[ f_x(i) = f(i, x) . \]

Now BP for $f$ gives BP for $f_x$ for every $x$.

\[ \# \text{ x's } = 2^{n-k} \leq 2^n \Rightarrow BP: \text{size}(f) \geq \lceil \frac{5 \times 3}{4} \rceil \]
\[ \geq \frac{n}{\log n} \]

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

Idea 2: for \( S \subseteq [n] \)

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

Above proof actually implies

\[ \text{BP-size}_{\{1,...,k\}}(f) \geq \frac{n}{\log n} \]

\[ \uparrow \]

\[ \log n \text{ variables} \]
Can we repeat this for other blocks.

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

\[ \begin{align*}
    s_1 & \quad s_2 & \quad \ldots & \quad s_k \\
    u_1 & \quad u_2 & \quad \ldots & \quad u_k
\end{align*} \]

**Function:** \( \text{DISTINCT?} (x_1, \ldots, x_k, y_1, \ldots, y_k) \)

\[ \text{DISTINCT?} (u_1, \ldots, u_k) = 1 \quad \text{if} \quad \forall i \neq j \quad u_i \neq u_j \]

\[ = 0 \quad \text{otherwise.} \]

**Claim:** \( \forall i, \# \text{ functions} \)

\[ \left\{ \begin{array}{c}
    f (u_i) = \text{DISTINCT}(a_1, \ldots, u_i, \ldots, a_k) \\
    a_1, \ldots, a_k
\end{array} \right\} \geq \left( \frac{2^k}{l} \right)^l \]

\( \geq \left( \frac{2^k}{l} \right)^l \)
Claim: \( \text{BP-size}_{\text{Distinct}}(D) \geq \frac{n}{\log n} \)

Claim: \( \text{BP-size}(f) \geq \sum_i \text{BP-size}_{s_i}(f) \)

Putting Claims Together: get

\[ \text{BP-size (Distinctness)} \geq \ell^2 k - \ell^2 \log \ell \]

Letting \( k = 2\log \ell \) and \( N = \ell^k \)

get \( L \geq a \left( \frac{n^2}{\log^2 n} \right) \)

\( \boxed{\text{NECIPORUK}} \)
**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 $\leq 3$ BP’s of exponential size...

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

Natural candidate: $\text{MAJORITY}(x_1, \ldots, x_n) = 1$ $\iff$ $\exists x: x \geq \frac{n}{2}$.
Barrington's Theorem:

If $f$ has $O(n)$ width poly size BP

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

(other bases similar)
In above

\[ f_i = 1 \text{ if top b.p. reaches } i^{th} \text{ state in middle level} \]

\[ g_i = 1 \text{ if bottom b.p. accepts starting at middle level.} \]

\[ \Rightarrow \text{ Much harder (unexpected)} \]

Ben Or + Cleve's proof: Simple idea.

"Strong Induction"
Register Machines

Given by $l$ registers $R_1 - R_l$

$S$ instructions

$I_1$

$I_2$

\vdots

$I_S$

$I_j$: of the form $R_i \leftarrow R_j + R_k \ast R_l$

or $R_i \leftarrow R_j + x \ast R_l$
Register Machine computes $f(x_1, \ldots, x_n)$ if it maps 

$$(R_0, \ldots, R_e) \rightarrow (R_0, \ldots, R_{e-1}, R_e + f(), R_e)$$

Strong Hypothesis: if $f$ has depth $d$

$\exists$ $\mathrm{AND, NOT}$ formula $1$st

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

Prop. $f$ has size $S$, $l$-Register m/c

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

($8$ in our case)
Proof:

\[ f \text{ computed by } M_i = \overline{I}_i \]

\[ \Rightarrow (1-f) \text{ computed by } \overline{I}_s \]

\[ \overline{I}_s' \]

\[ \overline{I}_s' = \overline{I}_s \text{ replace + by } - \]

\[ \overline{I}_{s+1} = R_e \leftarrow R_e + R_i \]

(Previously, \( R_e \) had \( R_e^0 - f R_i^0 \) & \( R_i \) \( R_i^0 \))

Now: \( R_e \leftarrow R_e^0 + (1-f) R_i^0 \)
Interesting one

\[ f = f_1 \land f_2 \]

\[ R_1 \]

\[ R_2 \]

\[ R_3 + f_1 \cdot R_1 \]

\[ R_2 + f_2 R_3 + f_1 f_2 R_1 \]

\[ R_3 + f R_1 \]

\[ R_2 + f_2 R_3 + f_1 f_2 R_1 \]

\[ R_1 \]

\[ R_2 + f_2 R_3 + f_1 f_2 R_1 \]

\[ R_3 \]