CSS.203.1 Computational Complexity Lecture 16.

Recap:
- $L \subseteq NL \subseteq P$
- $NL \subseteq L^2$
- Catalytic computation.

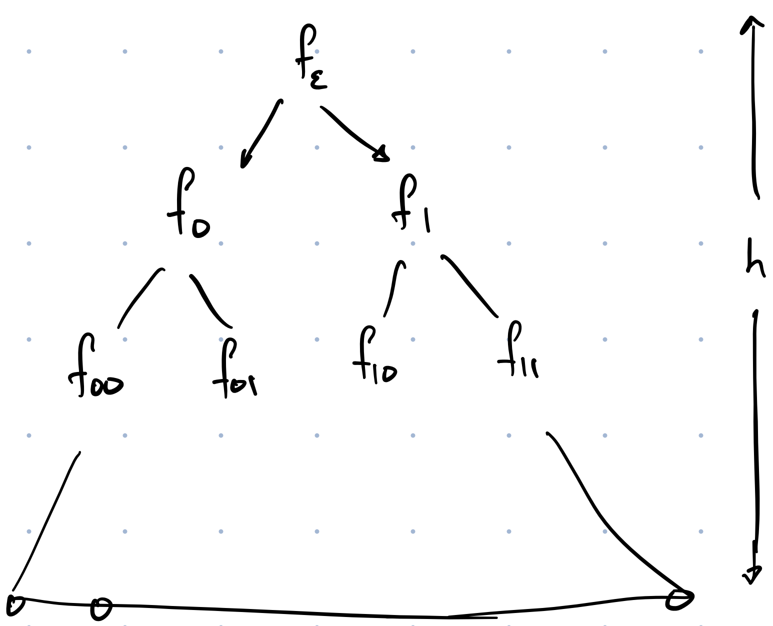
$$\begin{array}{c} CL \\ \uparrow \\ LogCFL \\ \uparrow \\ NL \end{array}$$

Belief: $L \subsetneq P$

Is there a natural candidate problem to separate?

Tree-Evaluation-Problem ($TEP_{h,k}$):

[Cook, McKenzie, Wehr, Braverman, Santhanam 2012]



For each $u \in \{0,1\}^h$, $v_u \in \{0,1\}^k$

For each $u \in \{0,1\}^{<h}$,
$f_u : \{0,1\}^k \times \{0,1\}^k \to \{0,1\}^k$.

Given all of the above compute the root value.

Input size: $2^h \cdot k + (2^h - 1)\, 2^{2k} \cdot k = 2^{O(k+h)}$

$h, k = O(\log n)$

Obs: $TEP_{h,k} \in P$.

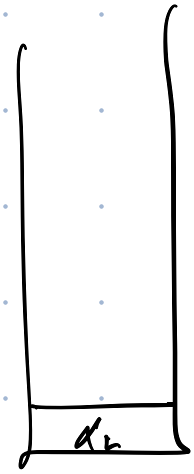
Naive low-space algo

$TEP_{h,k}$:

$\alpha_L = TEP_{h-1,k}$ (left subtree)

$\alpha_R = TEP_{h-1,k}$ (right subtree)

return $f_{root}(\alpha_L, \alpha_R)$



$$\begin{aligned} Space(h,k) &= Space(h-1,k) + O(k) \\ &= O(hk) \\ &= O(\log^2 n) \end{aligned}$$

[Cook-Mertz 2021] $TEP \in DSPACE(\log^2 n / \log\log n)$

[Cook-Mertz 2024] $TEP \in DSPACE(\log n \cdot \log\log n)$.

Thm [Cook-Mertz] $TEP_{h,k} \in DSPACE((h+k)\log k)$

Key: do the naive algo catalytically.

$TEP_{h,k}$:

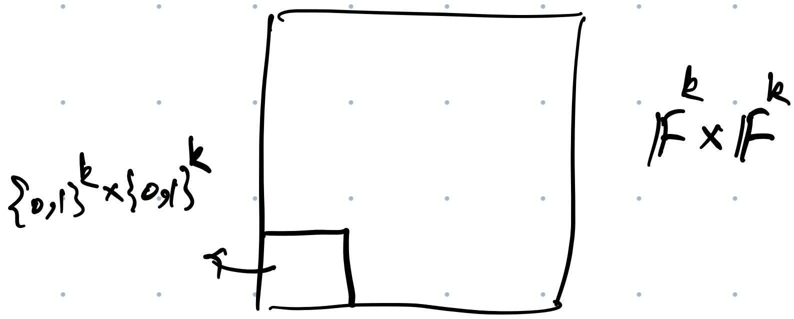
$\alpha_L = TEP_{h-1,k}$ (left)

$\alpha_R = TEP_{h-1,k}$ (right)

$\beta = f_u(\alpha_L, \alpha_R)$

Aside: Polynomials, extensions & interpolation.

$f: \{0,1\}^k \times \{0,1\}^k \to \{0,1\}^k \qquad \{0,1\} \subseteq \mathbb{F}$



Extension to $\mathbb{F}$:

$$F^{(i)}(x_1,..,x_k,y_1,..,y_k) \in \mathbb{F}[x_1,..,x_k,y_1,..,y_k]$$

s.t

$$F^{(i)}(a_1,..,a_k,b_1,..,b_k) = f^{(i)}(a_1,..,a_k,b_1,..,b_k)$$

wherever $a_1,..,a_k,b_1,..,b_k \in \{0,1\}$.

$$F^{(i)}(x_1,..,x_k,y_1,..,y_k) = \sum_{\substack{a_1..a_k \\ b_1..b_k \in \{0,1\}}} f^{(i)}(\bar{a},\bar{b}) \cdot \delta_{\bar{a},\bar{b}}(\bar{x},\bar{y})$$

where

$$\delta_{\bar{a},\bar{b}}(\bar{x},\bar{y}) = \prod_{i=1}^{k} \big(a_i x_i + (1-a_i)(1-x_i)\big) \cdot \big(b_i y_i + (1-b_i)(1-y_i)\big)$$

$\deg F^{(i)} \leq 2k.$

Obs: Given $f : \{0,1\}^k \times \{0,1\}^k \to \{0,1\}^k$, and $\alpha_1, \ldots, \alpha_k, \beta_1, \ldots, \beta_k \in \mathbb{F}$, we can compute $F^{(i)}(\alpha_1, \ldots, \alpha_k, \beta_1, \ldots, \beta_k)$ using space $O(k + \log |\mathbb{F}|)$

Pf: Just sum term by term.

$$F^{(i)}(\bar{\alpha}, \beta) = \sum_{\bar{a}, b \in \{0,1\}^k} f(\bar{a}, b) \cdot \delta_{a,b}(\alpha, \beta)$$

$\square$

$$\underbrace{\left\{ f_u : \{0,1\}^k \times \{0,1\}^k \to \{0,1\}^k \right\}}_{TEP} \xrightarrow[|\mathbb{F}| \approx 2k]{O(k)} \underbrace{\left\{ F_u : \mathbb{F}^k \times \mathbb{F}^k \to \mathbb{F}^k \right\}}_{TEP^*_{\mathbb{F}}}$$

$$\underbrace{\phantom{TEP^*_{\mathbb{F}}}}_{L}$$

$$\text{Space}(TEP) \leq \text{Space}(TEP^*_{\mathbb{F}}) + O(k) + \log L$$

Interpolation: $g(x) = g_0 + g_1 x + \cdots + g_{d-1} x^{d-1} \in \mathbb{F}[x]$

Suppose $\alpha_1, \ldots, \alpha_d$ are distinct elements in $\mathbb{F}$, and if we are given $g(\alpha_1), \ldots, g(\alpha_d)$, then we know g

$$\begin{bmatrix} 1 & \alpha_1 & \alpha_1^2 & \cdots & \alpha_1^{d-1} \\ 1 & \alpha_2 & \alpha_2^2 & \cdots & \alpha_2^{d-1} \\ & & \vdots & & \\ 1 & \alpha_d & \alpha_d^2 & \cdots & \alpha_d^{d-1} \end{bmatrix} \begin{bmatrix} g_0 \\ \vdots \\ g_{d-1} \end{bmatrix} = \begin{bmatrix} g(\alpha_1) \\ \vdots \\ g(\alpha_d) \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} g_0 \\ \vdots \\ g_{d-1} \end{bmatrix} = \begin{bmatrix} \quad V^{-1} \quad \end{bmatrix} \begin{bmatrix} g(\alpha_1) \\ \vdots \\ g(\alpha_d) \end{bmatrix}$$

In particular, $g_0 = g(0) = \sum_{i=1}^{d} \gamma_i \, g(\alpha_i)$

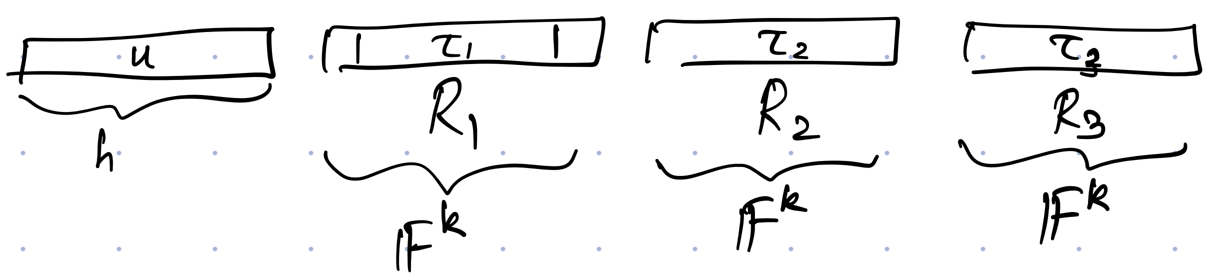
Fact: There are "nice" choices of α_i's that make the γ_i's computable in low-space.

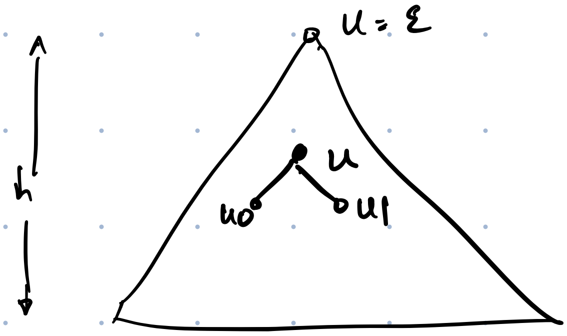
The Cook-Mertz Algorithm: $TEP^*_{\mathbb{F}}$

$$\left\{ F_u^{(i)} : \mathbb{F}^k \times \mathbb{F}^k \longrightarrow \mathbb{F} \right\}_{\substack{u \in \{0,1\}^{<h} \\ i \in [k]}} , \quad \left\{ v_u \in \{0,1\}^k \right\}_{u \in \{0,1\}^h}$$

$\deg F_u^{(i)} \leq 2k$. $\qquad |\mathbb{F}| \gtrsim 2k$

Program memory model:





For $u \in \{0,1\}^h$, V_u = leaf value.

For $u \in \{0,1\}^{<h}$:

$$V_u = f_u(V_{u0}, V_{u1})$$

Inductive goal: Build P_u that does

u	τ_1	τ_2	τ_3

$\downarrow$ $\quad \forall \tau_1, \tau_2, \tau_3$

u	τ_1	τ_2	$\tau_3 + V_u$

Inductive assumption, we have P_{u0} & P_{u1} with us

τ_1	τ_2	τ_3

$\downarrow P_{u0}$

$\tau_1 + V_{u0}$	τ_2	τ_3

τ_1	τ_2	τ_3

$\downarrow P_{u1}$

τ_1	$\tau_2 + V_{u1}$	τ_3

Attempt 1:

- Run P_{u0}
- Run P_{u1}
- $R_3^{(i)} \leftarrow R_3 + F_u^{(i)}(R_1, R_2)$ $\quad i = 1, \ldots, k.$
- Run P_{u0}^{-1}
- Run P_{u1}^{-1}

But we end with $\tau_3 + F_u(\tau_1 + V_{u0}, \tau_2 + V_{u1})$ in R_3 whereas we wanted $\tau_3 + F_u(V_{u0}, V_{u1})$

Attempt 2:

- $\triangleright$ Scale R_1 & R_2 by $\alpha \in \mathbb{F} \setminus \{0\}$.
- $\triangleright$ Run P_1 & P_2
- $\triangleright$ $R_3^{(i)} \leftarrow R_3^{(i)} + F_u^{(i)}(R_1, R_2)$
- $\triangleright$ Run P_1^{-1}, P_2^{-1}
- $\triangleright$ Scale R_1 & R_2 by α^{-1}.

Now, R_3 holds $\tau_3 + F_u(\alpha\tau_1 + V_{u0}, \alpha\tau_2 + V_{u1})$

$$G(t) = F_u(t \cdot \tau_1 + V_{u0}, t \cdot \tau_2 + V_{u1})$$

$\hookrightarrow$ deg $\leq 2k$

$$\therefore G(0) = \sum_{i=0}^{2k} \gamma_i \, G(\alpha_i)$$

$$= \sum \gamma_i \cdot F_u(\alpha_i \tau_1 + V_{u0}, \alpha_i \tau_2 + V_{u1})$$

Interpolation

P_u: For $j = 1, \ldots, 2k+1$:

$$R_1 \leftarrow \alpha_j R_1$$
$$R_2 \leftarrow \alpha_j R_2$$
Run P_{u0}, P_{u1}
$$R_3^{(i)} \leftarrow R_3^{(i)} + \gamma_j \cdot F_u^{(i)}(R_1, R_2) \quad i = 1, \ldots, k.$$
Run P_{u0}^{-1}, P_{u1}^{-1}
$$R_1 \leftarrow R_1 \cdot \alpha_j^{-1}$$
$$R_2 \leftarrow R_2 \cdot \alpha_j^{-1}$$

At the end, R_3 holds $\displaystyle \tau_3 + \sum_{j=1}^{2k+1} \gamma_j F_u(\alpha_j \tau_1 + v_{u0}, \alpha_j \tau_2 + v_{u1})$

$$= \tau_3 + F_u(v_{u0}, v_{u1})$$

Local space: storing i, j $= O(\log k)$

$\therefore$ $\mathrm{LocalSpace}(P_u) = \mathrm{LocalSpace}(P_{ub}) + O(\log k)$.

$\therefore$ Overall space $= O(h + k\log k + h \log k)$
$$= O((h+k)\log k)$$
$$= O(\log n \cdot \log\log n)$$

[Goldreich]: $O(k + h \log k)$

Choose $|\mathbb{F}| = q$ so that $2^k \approx q^q$

$q \approx \frac{k}{\log k}$

$$\underbrace{\boxed{}}_{\{0,1\}^k \text{ or } \mathbb{F}^{|\mathbb{F}|}}$$

Can do multivariate interpolation

Global space $= 3k + h$

Local space $= O(\log q)$ per recursion depth

$= O(h \cdot \log k)$ overall.

TEP with fan-in d: $O(dk + h \log dk)$

Next class: Ryan Williams' result.

1 INTRODUCTION (Cook-Mertz)

In complexity theory, many fundamental questions about time and space remain open, including their relationship to one another. We know that TIME(t) is sandwiched between SPACE$(\log t)$ and SPACE$(t/\log t)$ [18], and both containments are widely considered to be strict, but we have made little progress in proving this fact for any t.