CSS.203.1 Computational Complexity. Lec 15.

Agenda for the next three lectures

- ▷ Ryan Williams' breakthrough:

$$\mathrm{DTIME}(t(n)) \subseteq \mathrm{DSPACE}\left(\sqrt{t(n)\log t(n)}\right)$$

- ▷ Cook-Mertz Theorem:

$$\text{Tree Evaluation Problem} \in \mathrm{DSPACE}(\log n \log\log n)$$

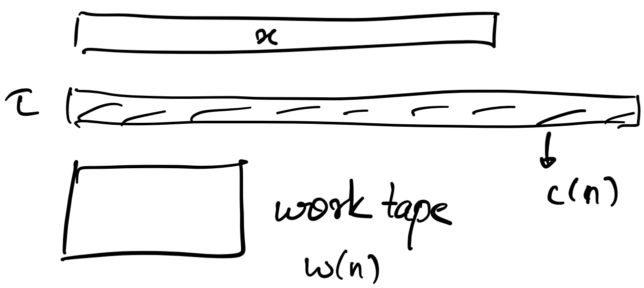
- ▷ Catalytic computation. (Buhrman, Cleve, Koucky, Loff, Speelman)

Recap:

$$L \subseteq NL \subseteq P \qquad\qquad NL \subseteq L^2.$$
$$NL = coNL$$

Machine model: read-only input tape + work tape(s)

What is catalytic computation?



Restriction:

At the end, catalytic tape contents <u>must</u> be restored!

Defn: CSPACE$(w(n), c(n))$: Languages that are computable by catalytic machines with catalytic tape size $c(n)$ & work tape $w(n)$.

Convention: $\mathrm{CSPACE}(w(n)) := \mathrm{CSPACE}(w(n), 2^{O(w(n))})$

Obs: $\mathsf{L} \subseteq \mathsf{CL}$

Qn: Is the containment strict?
Can CL do tasks beyond L? Yes*

Reversible computation:

$R_i \in \mathbb{R}$ (think $\mathbb{Z}$)

$$\boxed{R_1} \quad \boxed{R_2} \quad \boxed{} \quad \cdots \quad \boxed{R_m}$$

Instructions: $\quad R_i \leftarrow R_i \pm u_j u_k$

u_j : constant input

u_k : another register

Eg:

$$\begin{array}{lcl}
R_1 \leftarrow R_1 + x & & R_2 \leftarrow R_2 - R_1 z \\
R_2 \leftarrow R_2 + R_1 R_3 & \xrightarrow{\text{inverse}} & R_1 \leftarrow R_1 + \gamma R_2 \\
R_1 \leftarrow R_1 - \gamma R_2 & & R_2 \leftarrow R_2 - R_1 R_3 \\
R_2 \leftarrow R_2 + R_1 z & & R_1 \leftarrow R_1 - x
\end{array}$$

Transparent computation: A reversible program P transparently computes $f(x_1, \ldots, x_n)$ on register R_1 if $\forall\ I_1, \ldots, I_m$

$$\boxed{I_1} \ \cdots \ \boxed{I_m}$$

$$\big\downarrow P$$

$$\boxed{I_1'} \ \boxed{I_2'} \ \cdots \ \boxed{I_m'}$$

with $I_1' = I_1 + f(\bar{x})$.

Clean computation:
$$\tau_1' = \tau_1 + f(\bar{x})$$
$$\tau_2' = \tau_2$$
$$\vdots$$
$$\tau_m' = \tau_m .$$

Claim: Suppose there is an m-register program P that transparently computes $f(\bar{x})$, then there is an $(m+1)$-register program P' that cleanly computes $f(x)$, with $|P'| \leq 2|P| + 2$

Pf:

$$\boxed{\tau_0}\ \boxed{\tau_1}\ \cdots\ \boxed{\tau_m}$$

$R_0 \leftarrow R_0 - R_1$ $\qquad\qquad \tau_0 + f$

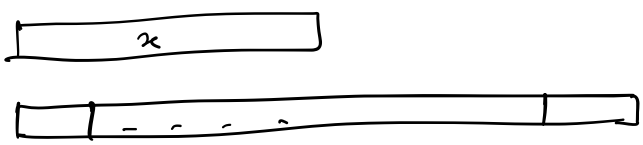
P $\qquad\qquad \tau_1$

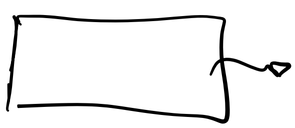
$R_0 \leftarrow R_0 + R_1$ $\qquad\qquad \tau_2$

P^{-1} $\qquad\qquad \tau_m$

$\square$.

Overview of catalytic computation:





storing some register value.
program counter etc

What can we compute transparently?

$$f(x_1, \dots, x_n) : \mathbb{R}^n \to \mathbb{R}$$

Lemma 1: Suppose $f(\bar{x})$ can be transparently computed by P_f, & $g(\bar{x})$ can be transparently computed by P_g then we can transparently compute $f+g$ by: $P = P_f^{\pi} \circ P_g^{\pi}$

Pf: Duh!

Lemma 2: If P_f cleanly computes f, & P_g cleanly computes g, then we can also cleanly compute fg.

Pf: $\sigma : R_1 \leftarrow R_1 - R_2 R_3$

$$P_f^{(2)} \; \sigma \; P_g^{(3)} \; \sigma^{-1} \; P_f^{(2)^{-1}} \; \sigma \; P_g^{(3)^{-1}} \; \sigma^{-1}$$

$\tau_1 - \cancel{I_2 I_3} - \cancel{f I_3} + \cancel{I_2 I_3} + \cancel{I_2 g} + \cancel{f I_3} + fg \cancel{- I_2 I_3} \cancel{- I_2 g} + \cancel{I_2 I_3}$

τ_2

τ_3

$\square$

Thm: [Ben-Or & Cleve] If f is computable by a a fan-in 2 depth d formula, then we can cleanly compute f via a program of length $\leq 4^d$ using 3-registers.

Fact: If f is computable by a formula of size s, then it is also computable by a formula of size poly(s) & depth $O(\log s)$ [Brent, Spira].

Obs: Ben-Or Cleve also works when R is not commutative.

Qn: Suppose we are able to find $k \times k$ matrices $M_1, \ldots, M_s$ with entries in R or x_i and suppose $(M_1 \cdots M_s)_{1,1} = f(\bar{x})$. Can we compute f cleanly?

$\boxed{R_1}$ $\boxed{R_2}$ $\boxed{R_3}$

Matrix registers.

$\hookrightarrow$ k^2 usual registers.

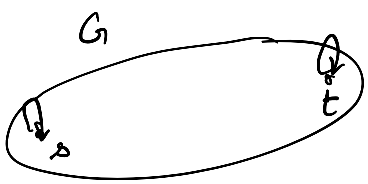
$R_1 \leftarrow R_1 - R_2 R_3$

$\square - \square\,\square \rightsquigarrow$ k^3 instr. in the base world.

Corollary: If $f(x) = (M_1 \cdots M_s)_{1,1}$ and each $M_i \in R^{k \times k}$, then f can be cleanly computed by a program of length $O(s^2 k^3)$ with $O(k^2)$ registers.

Thm: DirPath $\in$ CL $\quad (\Rightarrow \text{NL} \subseteq \text{CL})$

Pf:



$(A^n)_{s,t}$

Run the above algo.

Qn: What ring R do you work with?

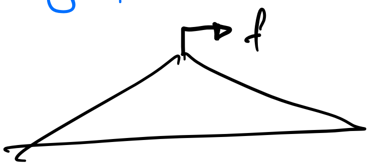
Fix: Try primes $p_1, p_2, \ldots, p_n$

If any p_i shows $(A^n)_{s,t} \neq 0 \bmod p_i$
return Yes
(after undoing)

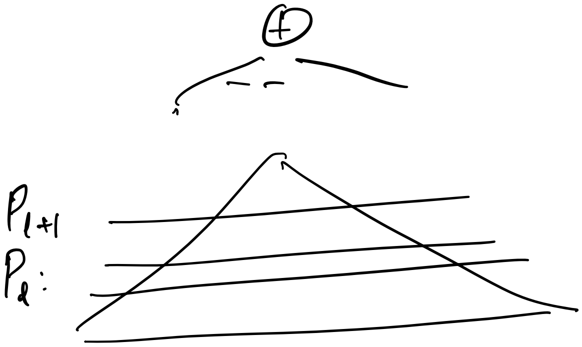
$\square$.

Infact gapL $\subseteq$ CL

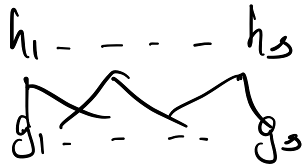
Going further: f - computable by a circuit of size s, depth d



size s
depth $d = O(\log s)$



Each gate g has a few registers $R_{g,0}, \ldots, R_{g,5}$.



Case 1: $h = g_1 + g_2 + \cdots + g_k$

$$R_{h,0} \leftarrow R_{h,0} - R_{g_1,0} - R_{g_2,0} - \cdots - R_{g_k,0}$$

$$P_l$$

$$R_{h,0} \leftarrow R_{h,0} + R_{g_1,0} + \cdots + R_{g_k,0}$$

$$P_l^{-1}.$$

Length: $2|P_l| + O(s)$

\# Registers: 1 for h.

Case 2: $h = g_1 \times g_2$

$$R_0 \leftarrow R_0 + R_1R_2 + R_1R_4 + R_3R_2$$

$$\left.\boxed{\begin{array}{l} R_1 \leftarrow R_1 + g_1 \\ R_2 \leftarrow R_2 + g_2 \end{array}}\right. P_\ell$$

$$R_3 \leftarrow R_3 + R_1$$

$$R_4 \leftarrow R_4 + R_2$$

$$R_0 \leftarrow R_0 + R_1R_2$$

$$\boxed{\begin{array}{l} R_2 \leftarrow R_2 - g_2 \\ R_1 \leftarrow R - g_1 \end{array}}$$

$$R_0 \leftarrow R_0 - R_1R_4 - R_3R_2$$

$$\begin{array}{ll} I_0 + \cancel{I_1I_2} + \cancel{I_1I_4} + \cancel{I_3I_2} & \\ \quad \cancel{I_1I_2} + \cancel{I_1g_2} + \cancel{I_2g_1} + g_1g_2 & \\ \quad -\cancel{I_1I_4} - \cancel{I_1I_2} - \cancel{I_1g_2} & \\ \quad - \cancel{I_3I_2} - \cancel{I_1I_2} - \cancel{g_1I_2} & \end{array}$$

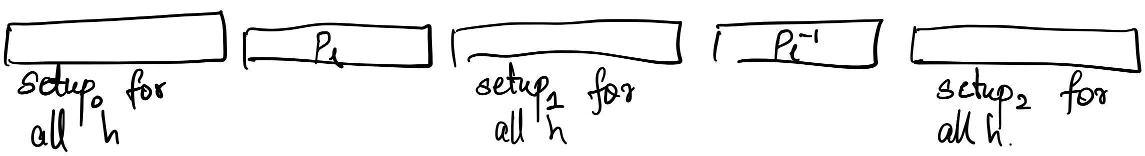
$$I_1$$

$$I_2$$

$$I_3 + I_1 + g_1$$

$$I_4 + I_2 + g_2$$

Length of program: $2|P_\ell| + 8$, # registers for h: 3



Total length $\le 2|P_\ell| + \mathrm{poly}(s)$

$\therefore$ Final length $= 2^d \cdot \mathrm{poly}(s)$. #registers : $O(s)$. $\square$

Thm: LOGCFL $\subseteq$ CL.

Thm: [BCKLS] uniform $TC^1 \subseteq$ CL.

How large is CL? $\quad CL \subseteq PSPACE$

Thm: [BCKLS] $\quad CL \subseteq ZPP$.

Next time: TEP