Matrix Rigidity

Hierarchy Theorems

Sasha Golovnev

October 19, 2020

Overview

- Recall that we want $M \in \mathbb{F}_2^{n \times n}$, $M \in \mathbf{E}^{\mathbf{NP}}$

$$\mathcal{R}_M^{\mathbb{F}_2}(\varepsilon n) \geq \Omega(n^{1+\delta}) .$$

- We'll prove that there is $M \in \mathbb{F}_2^{n \times n}$, $M \in \mathbf{P}^{\mathbf{NP}}$

$$\mathcal{R}_M^{\mathbb{F}_2}(2^{\log n / \log\log n}) \geq \Omega(n^2) .$$

- We'll use
  - Orthogonal Vectors
  - Non-deterministic Hierarchy Theorem
  - Rectangular PCPs

Hierarchy Theorems

✓ $\mathrm{DTime}[f(n)] \subsetneq \mathrm{DTime}[g(n)]$ Time H.T.

$g(n) >> f(n)$

$\mathrm{DSpace}[f(n)] \subsetneq \mathrm{DSpace}[g(n)]$ Space H.T.

✓ $\mathrm{NTime}[f(n)] \subsetneq \mathrm{NTime}[g(n)]$ NTime H.T.

$\mathrm{NSpace}[f(n)] \subsetneq \mathrm{NSpace}[g(n)]$ NSpace H.T.

Circuit-Size$[f(n)] \subsetneq$ Circuit-Size$[g(n)]$ Circuit H.T.

Data-Structures H.T.

Williams' approach:

Assume to the contrary that

- $NTime[2^n]$ can be computed by small circuits

By NTime Hier T:

- $\exists L \in NTime[2^n] \setminus NTime[2^n/n]$

- Use the fact that L has a small circuits, come up with genius alg, to solve language L in $NTime[2^n/n]$.

Deterministic Hierarchy Theorem

DTime[f(n)] – all languages that can be computed by TM of size $O(f(n))$

Theorem (HS65)

If f, g are time constructible and $f(n) \cdot \log f(n) = o(g(n))$, then

$$\mathsf{DTIME}[f(n)] \subsetneq \mathsf{DTIME}[g(n)] .$$

f(n) is time constructible if one can compute f(n) in time f(n).

$n \to$ [$n^3, 2^n, n!, n^{100}/\log\log n$] — can compute $n^3, 2^n, n!, \ldots$

Theorem (HS65)

If f, g are time constructible and $f(n) \cdot \log f(n) = o(g(n))$, then

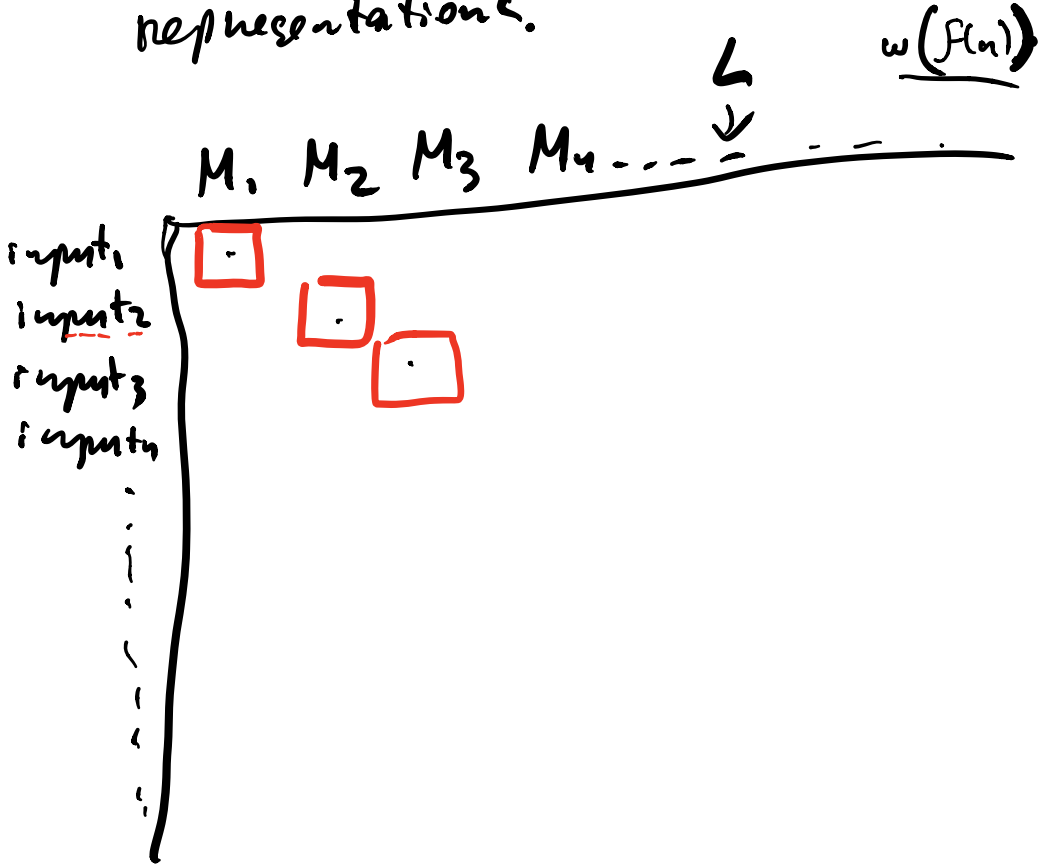
$$\mathsf{DTIME}[f(n)] \subsetneq \mathsf{DTIME}[g(n)] .$$

Diagonalization

Enumerate all TM.

$\{0,1\}^* \Longleftrightarrow$ TM

Every TM has infinitely many representations.



TM M:

On input i, Run

$M_i(input_i)$ for $\geq f(n)$ steps.

- if it outputs 0 $\Rightarrow$ I output 1
- if it outputs 1 $\Rightarrow$ I output 0
- if it doesn't output $\Rightarrow$ I output 1.

M also defines language L.

$$L \in DTime[g(n)] \setminus DTime[f(n)].$$

I. $L \notin DTime[f(n)]$. $n = |input_i|$

Assume TM P that runs in time $f(n)$ and decides P.

P = M:

$P(input_i)$ it runs for $f(n)$ steps outputs $b \in \{0,1\}$.

$M(input_i)$ – runs $M_i(input_i) = P(input_i)$

$1-b \in \{0,1\}$.

$$P(input_i) \neq M(input_i)$$

DTime[f(n)] – Machines in time $O(f(n))$

Say, P runs in time $10n^2$

Simulate for $n^2 \log\log n$ steps

$P(input_i) = M(input_i)$ for small i.

Every machine (incl. P) has infinitely many representations $\Rightarrow$

$P = M_i$ for infinitely many is.

Π. $L \in DTime[g(n)]$

I need **one** TM that simulates **any** TM for $f(n)$ steps.

Universal Turing Machine

∃ UTM that runs any machine for $f(n)$ steps, takes time

$O(f(n) \cdot \log f(n))$

By simulating any k-tape TM on a 2-tape TM with a log-loss in runtime. But then sim two-tape essentially without any loss.

My language can be solved in time $\Omega(f(n) \log f(n))$. $\Rightarrow$

$$L \in DTime[g(n)]$$

My TM has to have a counter (for $\approx f(n)$ steps).

$f(n)$ - time constructible.

Non-Deterministic Hierarchy Theorem

Ideal would say: $f(n) = o(g(n))$

$\text{TIME}(f(n)) \subsetneq \text{Time}[g(n)]$

Theorem (C72,Z83)

If f, g are time constructible and $f(n+1) = o(g(n))$, then

$$\text{NTIME}[f(n)] \subsetneq \text{NTIME}[g(n)] .$$

$1^n = 1 \ldots 1$ (n times)

unary languages

$L \subseteq \{1\}^*$

Universal Non-Deterministic TM
Can running $O(f(n))$

Previous simple diag. works for HM:

- DSpace HT.
- NSpace HT (+Immerman Thm)

doesn't work NTime; because

NTime is not (known to be) closed under complement.

I cannot take ND TM on input and output the opposite.

Instead of flipping the answer of M_i on input i, I'll flip on some input from exp large interval.

I partition integers.

$[\ell_1, u_1], [\ell_2, u_2], [\ell_3, u_3], \ldots$

$\ell_1 = 1 \qquad \ell_2 = u_1 + 1 \qquad \ell_3 = u_2 + 1$

$u_i >> 2^{\ell_i}$

input$_1$ ... input$_{10}$ input$_{11}$ ... input$_{1000}$

$[1, 10], [11, 1000], [1001, \ldots], \ldots$

M_j language L will differ from M_i somewhere in the interval i.

M_j language L (define via NTM):

on input: $i \in [\ell_j, u_j]$ (working w int j; I want to fool M_j)

If $i < u_j$:
Running $M_j(\text{input}_{i+1})$ ...

If $i = u_j$:
Running $M_j(\text{input}_{\ell_j})$, ✓
output the opposite.

I. $L \in NTime[g(n)]$.

1. F is g time constructible, so I can clock off F
2. I have UTM runs in time $F(n+1) = o(g(n))$
3. input of length $[|M_j| >> 2^{F(\ell_j)}]$

$M_j(input_j)$

Brute force non-determinism $2^{F(\ell_j)}$ $\Rightarrow$ compute it in $DTime[2^{F(\ell_j)}]$

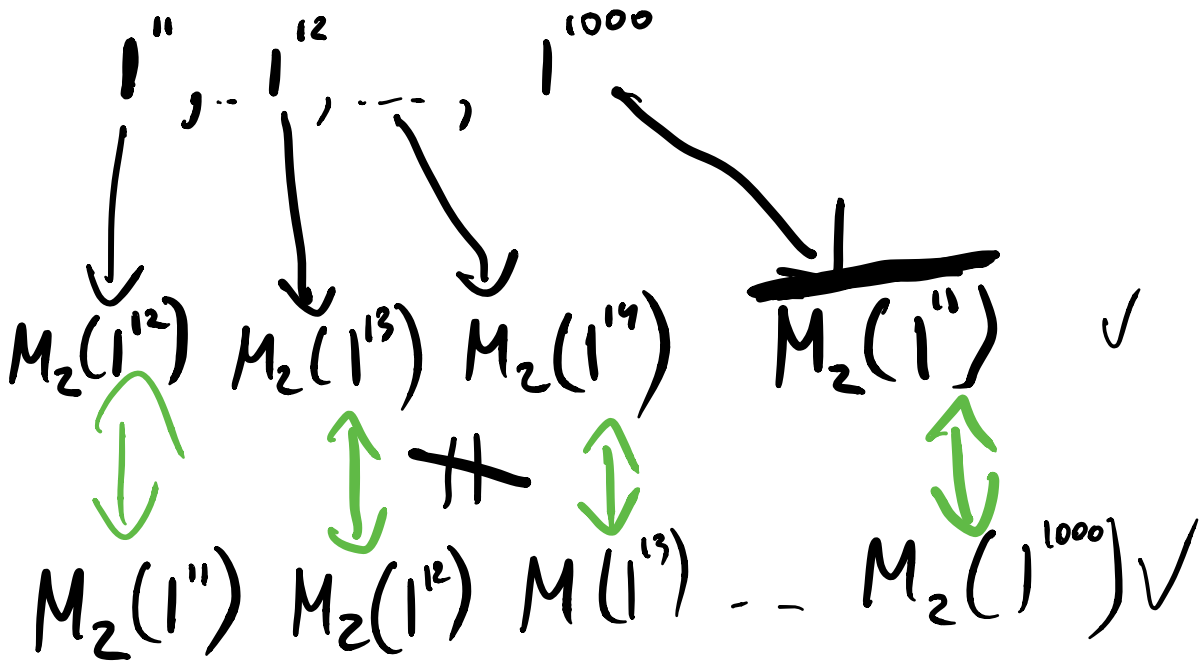
output the opposite.

II. $L \in NTime[F(n)]$.

$[1,10]$, $[11, 1000]$, ---

I fool TM M_2 on one of the inputs

$\{1^{11}, 1^{12}, \ldots, 1^{1000}\}$



Assume my machine & M_2 have same outputs on all these inputs:

$$M_2(1^{11}) = M_2(1^{12}) = M_2(1^{13}) = M_2(1^{14}) = \dots = M_2(1^{1000}) = M_2(1^{11})$$

$\Rightarrow L \notin NTime[f(n)]$ ☐

$$NTime[f(n)] \subseteq DTime[2^{f(n)}]$$

$$DTime[g(n)] = coDTime[g(n)]$$

Cor There exists an unary language $L \in NTime[2^n] \setminus NTime[2^n/n]$