CS5.203.1 Computational Complexity Lecture 17

Recap:
- ▷ $\mathrm{DTIME}(t) \subseteq \mathrm{DSPACE}(t)$
- ▷ $\mathrm{DSPACE}(t) \subseteq \mathrm{DTIME}(2^{O(t)})$
- ▷ $\mathrm{TEP}_{h,k,d} \in \mathrm{DSPACE}(kd + h\log kd)$

height, leaf label size, fan-in

Qn: Can we simulate $t(n)$-time with $o(t(n))$ space?

[Hopcroft-Paul-Valiant '75]: $\mathrm{DTIME}(t(n)) \subseteq \mathrm{DSPACE}\left(\frac{t(n)}{\log t(n)}\right)$

[Hopcroft-Ullman 68] For 1-tape TMs, $\mathrm{DTIME}_1(t(n)) \subseteq \mathrm{DSPACE}_1(\sqrt{t(n)})$

Is it possible that $\exists\, \varepsilon > 0$ with $\mathrm{DTIME}(t(n)) \subseteq \mathrm{DSPACE}(t(n)^{1-\varepsilon})$?

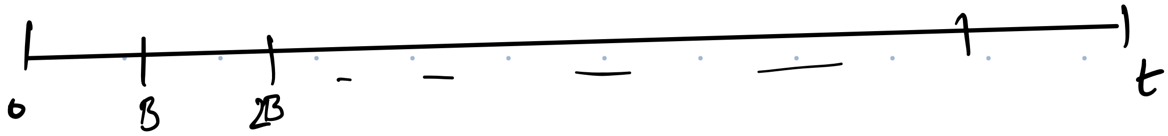
[Williams' 2025]: $\mathrm{DTIME}(t(n)) \subseteq \mathrm{DSPACE}\left(\sqrt{t(n)\log t(n)}\right)$

TLDR: reduce to TEP somehow, apply Cook-Mertz

Second, we find it fortunate that the main reduction of this paper (from time-t multitape Turing machine computations to TREE EVALUATION) was found *after* the Cook-Mertz procedure was discovered. Had our reduction been found first, the community (including the author) would have likely declared the following theorem (a cheeky alternative way of presenting the main reduction of Theorem 1.1) as a "barrier" to further progress on TREE EVALUATION, and possibly discouraged work on the subject:

> "**Theorem**." Unless the 50-year-old [HPV75] simulation $\mathsf{TIME}[t] \subseteq \mathsf{SPACE}[t/\log t]$ can be improved, TREE EVALUATION instances of constant arity, height h, and b-bit values cannot be solved in $o(h \cdot b/\log(h \cdot b))$ space.[9]

Overview (of HPV 75)



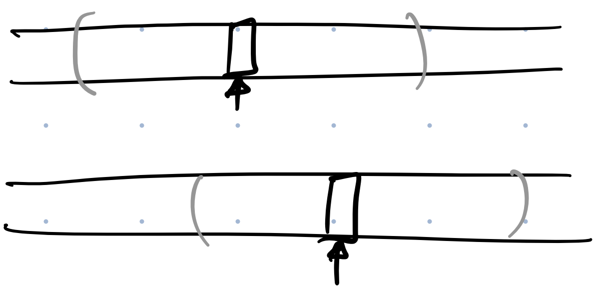
Configuration of a TM:

- state
- head positions
- tape contents

Qn: Suppose you are given the configuration at time step iB, how much space do you need to compute conf. at time step $(i+1).B$?

Just $O(B)$

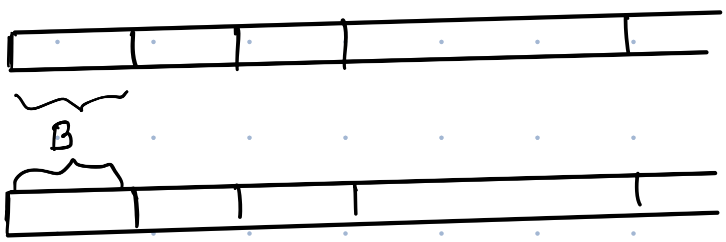
Qn: Do we need the entire config at time iB?



These are the only cells that may be affected in this epoch.

Given this "local config" we can simulate B steps and give the "new local config"

Splitting tapes into blocks.



Want the TM heads to switch across blocks only at time steps $B, 2B, 3B$.

Fact: TMs can be made block respecting with only const. factor increase in running time

7. **[Block-respecting TMs]** (10)

Hat-tip: This problem is from Ryan O'Donnell's complexity course

Let $B : \mathbb{N} \to \mathbb{N}$ be a "reasonable" function (increasing, time-constructible, yada yada). A deterministic Turing machine M is said to be *B-block-respecting* if it has the following property:

> Consider a length n input x. The tape(s) of the Turing machine are split into contiguous blocks of length $B(n)$ each. Each head of the Turing Machine crosses a block-boundary only at time-steps that are integer multiples of $B(n)$. (That is, within each block of $B(n)$ time steps, each head operates only within a particular block.)

Given a deterministic Turing machine M that runs in time $T(n)$, and a "reasonable" function $B : \mathbb{N} \to \mathbb{N}$, construct a deterministic B-block-respecting Turing machine M' for the same language while ensuring that the running time of M' is $O(T(n))$.

	siht	hasi	ftni	htro	orpe
this	isah	intf	orth	epro	blem
hasi	ftni	htro	orpe	melb	

[Hint: Suppose the machine M uses k tapes, you could perhaps make use of the above picture and use $3k$ tapes.]

TapeBlock(h, i): What block in tape h is the head in during epoch i?

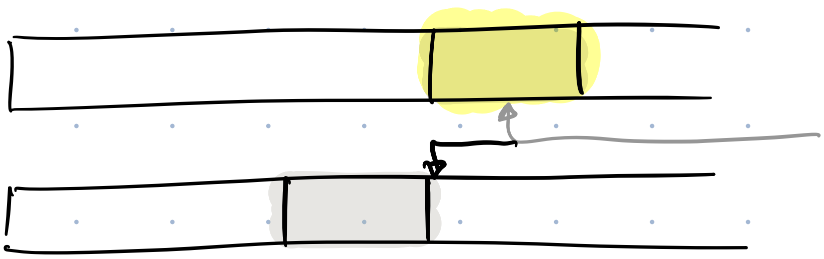
tape epoch

Contents(h,i): contents of above tape block after epoch i

LocalConfig(i) : State at the end of epoch i
Contents of TapeBlock$(*, i)$
+ head positions

$B' = O(B)$

What all do you need to compute LocalConfig(i)?



LocalConfig$(i-1)$

What if tape h changed block between epoch $i-1$ & epoch i?

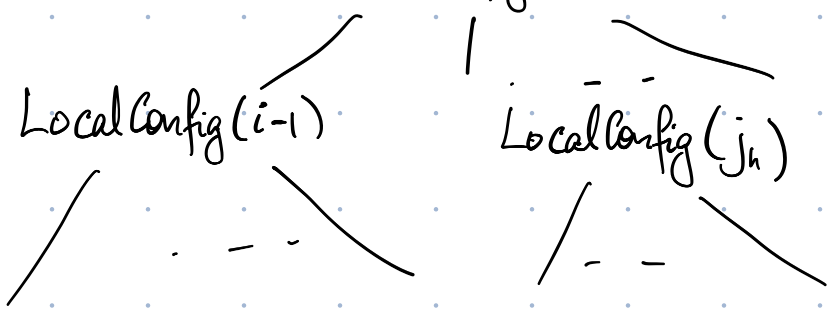
We need the contents after the last time we operated on this block..

$\therefore$ LocalConfig(i) depends on

- LocalConfig$(i-1)$
- LocalConfig$(j_h(i))$ $h \in$ [tapes]

where $j_h(i) = \max \left\{ j < i : \text{TapeBlock}(h, j) = \text{TapeBlock}(h, i) \right\}$

$$\text{LocalConfig}(i) \; : \; \{0,1\}^{B'(\text{tapes}+1)} \longrightarrow \{0,1\}^{B'}$$



This is basically a $TEP_{h,k,d}$ instance with

height $h = O(t/B)$
fan-in $d = \#\text{tapes} + 1$
leaf label size $k = B' = O(B)$

[Cook-Mertz] : $O(dk + h\log(kd))$ space

$$= O\left(B + \frac{t}{B}\log B\right)$$

$$= O(\sqrt{t\log t}) \qquad B = \sqrt{t\log t}$$

This gives us the required bound.

All of this depends on knowing the tree!

Qn: How do we compute this tree? We won't

Idea: Guess the tree, and verify with a bit more space.

$$m_{h,i} = \begin{cases} 1 & \text{head } h \text{ moves one block right in epoch } i \\ -1 & \text{head } h \text{ moves one block left in epoch } i \\ 0 & \text{head } h \text{ stays in same block in epoch } i \end{cases}$$

Qn: Given $\{ m_{h,i} : i \in [t/B],\ h \in [\text{tapes}] \}$ can you compute TapeBlock(h, i)?

$$\sum_{j<i} m_{h,j} + 1$$

This computation can be done with space $\log(t/B)$

Qn: Given $\mathcal{M} = \{ m_{h,i} : i \in [t/B], h \in [\text{tapes}] \}$
a time epoch i & tape block j on tape h,
what was the "last time" this tape block was
worked on ?

Just compute max(- - -)

Also computable from $\mathcal{M}$ using $O(\log(t/B))$ space.

How do we compute $\mathcal{M}$? You don't. You guess.

Fix a "guess" $\mathcal{M} = \{ m_{h,i} \}$.
and a "state sequence" $(q_1, .., q_{t/B}) \in Q^{t/B}$

Nodes = $\{ (h, i) : h \in [\text{tapes}], i \in [t/B] \}$
Values = contents of the tape block on tape h
after epoch i.

Leaves: $((h, i))$ where head h is accessing this
tape block for the first time.

Children of (h, i):

- $(h', i-1)$ for $h' \in$ [tapes]
- $(h', j_{h'})$ for $h' \in$ [tapes]

  $j_{h'}$ = "last access time"

  or this is a leaf.

Function at (h, i): Given contents of children.

- If any of them is "abort", return "abort"
- If state in $(h', i-1)$'s are inconsistent with q_{i-1}, return "abort".
- Simulate machine for B steps from these values.
- If state $\neq q_i$ or head movement$(h') \neq M_{h',i}$: return abort.

  Else:

  return tape contents after simulation.

Root of the tree: (output tape, t/B)

Obs: fn can be computed in space $O(B)$

Algo: For $M \in$ Possible Ms, $(q_1, \ldots, q_{t/B}) \in Q^{t/B}$:

Compute the root value for the TEP instance by using Cook-Mertz procedure on the implicit graph (space $(B + T/B)$ computable)

If answer = abort : go to next M, QSeq

Else : return $(q_{t/B} == \text{accept})$

Total space : $O(t/B) + O(B) + O(\log t/B)$

storing M, QSeq

computing leaf values, & node functions

computing "who are my children?"

$$+ O\left(B + \frac{t}{B} \log B\right)$$

[Cook-Mertz]

$$= O(B + t/B \log B)$$

$$= O(\sqrt{t \log t}) \quad \text{if } B = \sqrt{t \log t}$$

Open threads:

- Improve to $O(\sqrt{t})$? (Does this "need" proving TEP $\in$ L)?
- Extending it to RAM model.

- Does this extend to show $\mathrm{DTIME}(t) \subseteq \mathrm{ATIME}(t^{1-\varepsilon})$?
- Recursively using the space-efficient simulation?