

THE TREE EVALUATION PROBLEM

CONTEXT & RECENT RESULTS

IAN MERTZ

U. OF WARWICK

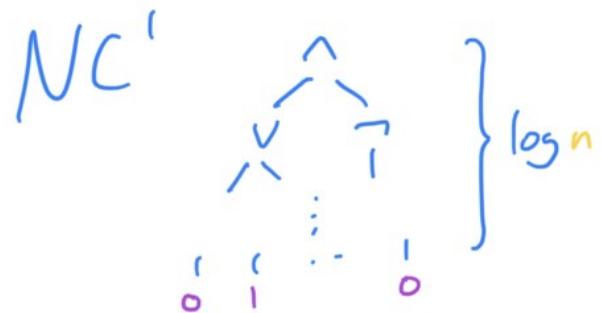
OCS, 2023.10.12

TREE EVALUATION

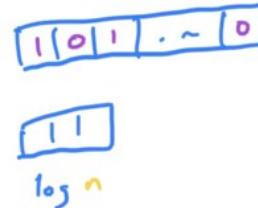
$$NC^1 \subseteq L \subseteq NL \subseteq NC^2 \subseteq P$$

TREE EVALUATION

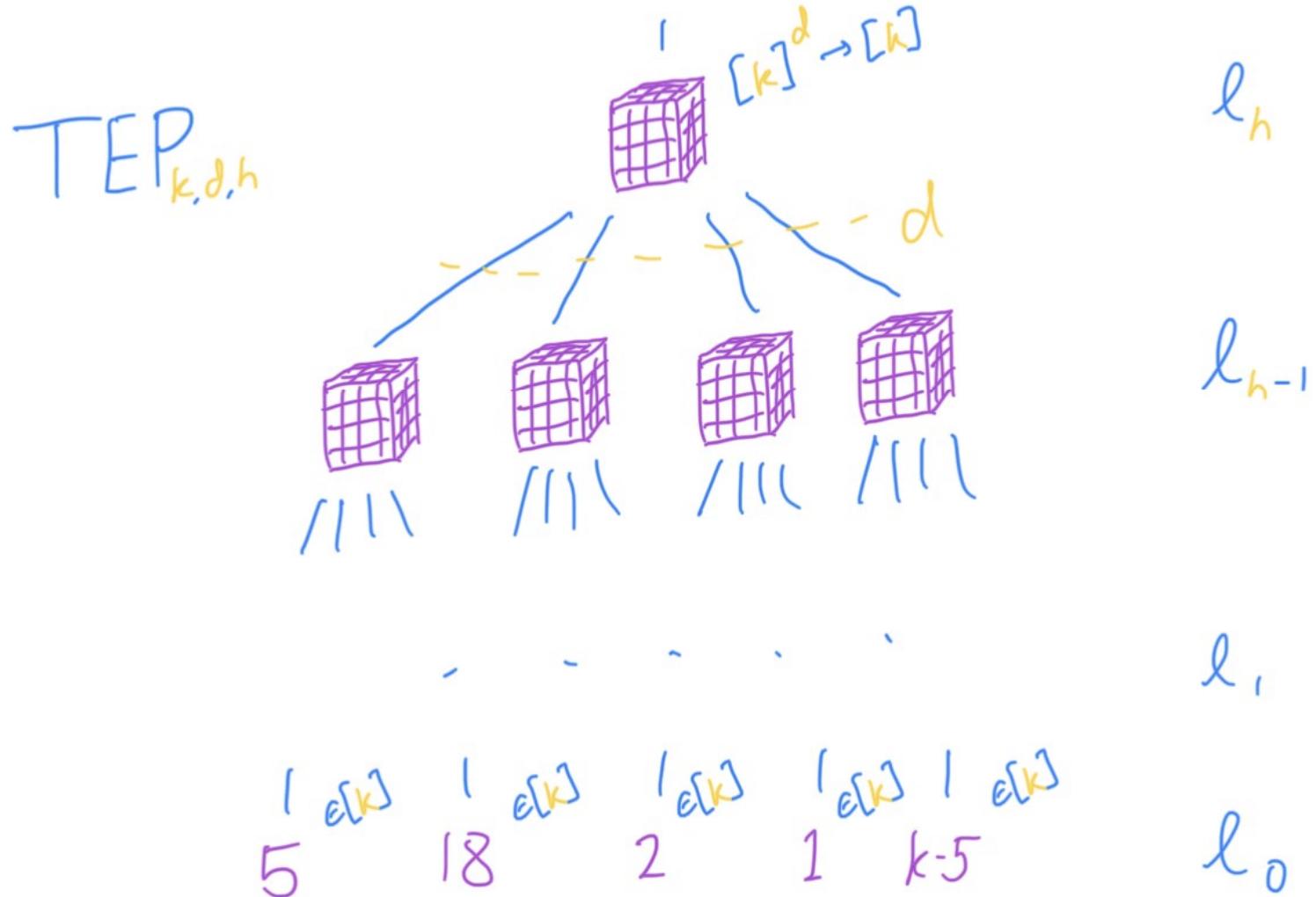
$$NC^1 \subseteq L \subseteq NL \subseteq NC^2 \subseteq P$$



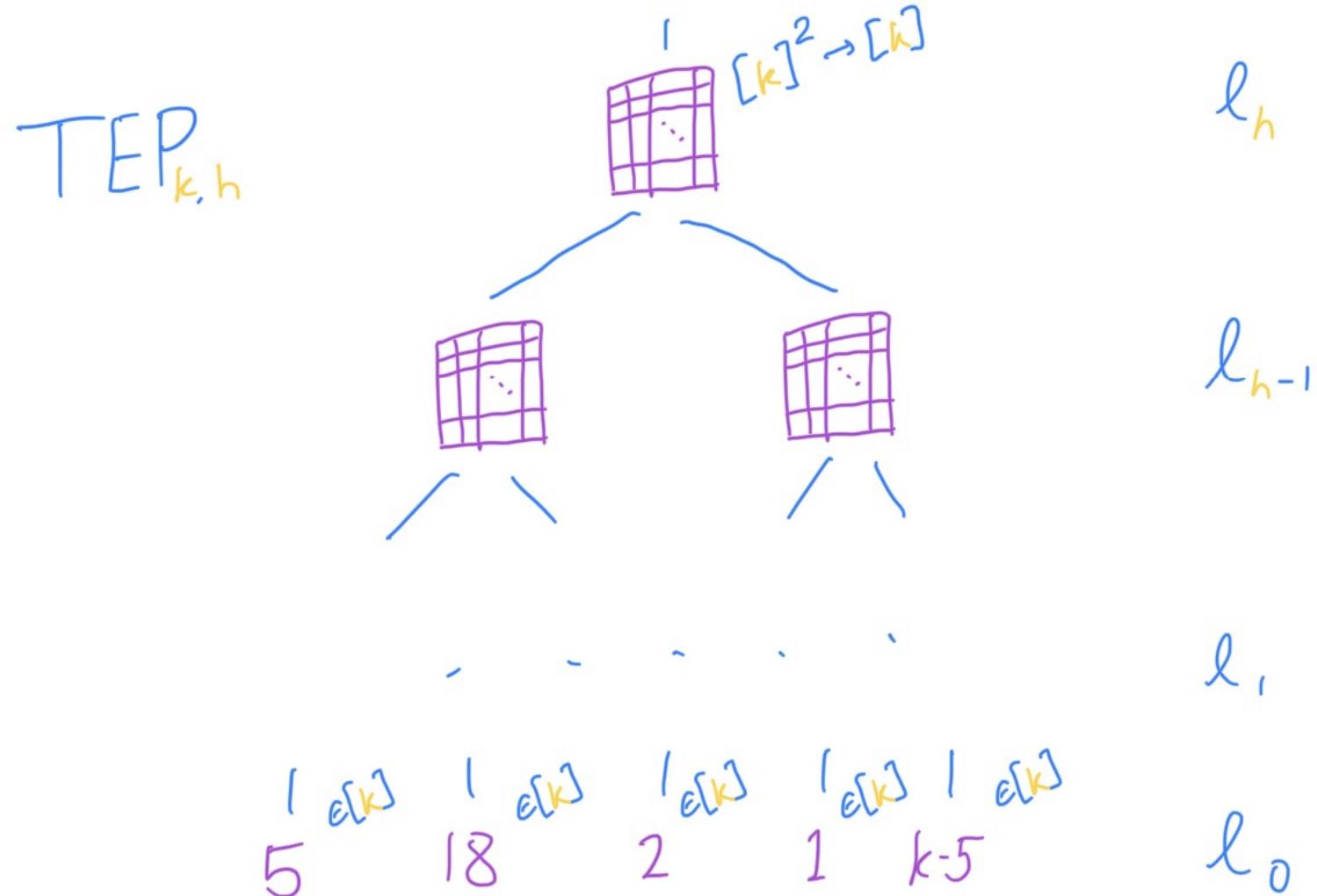
L



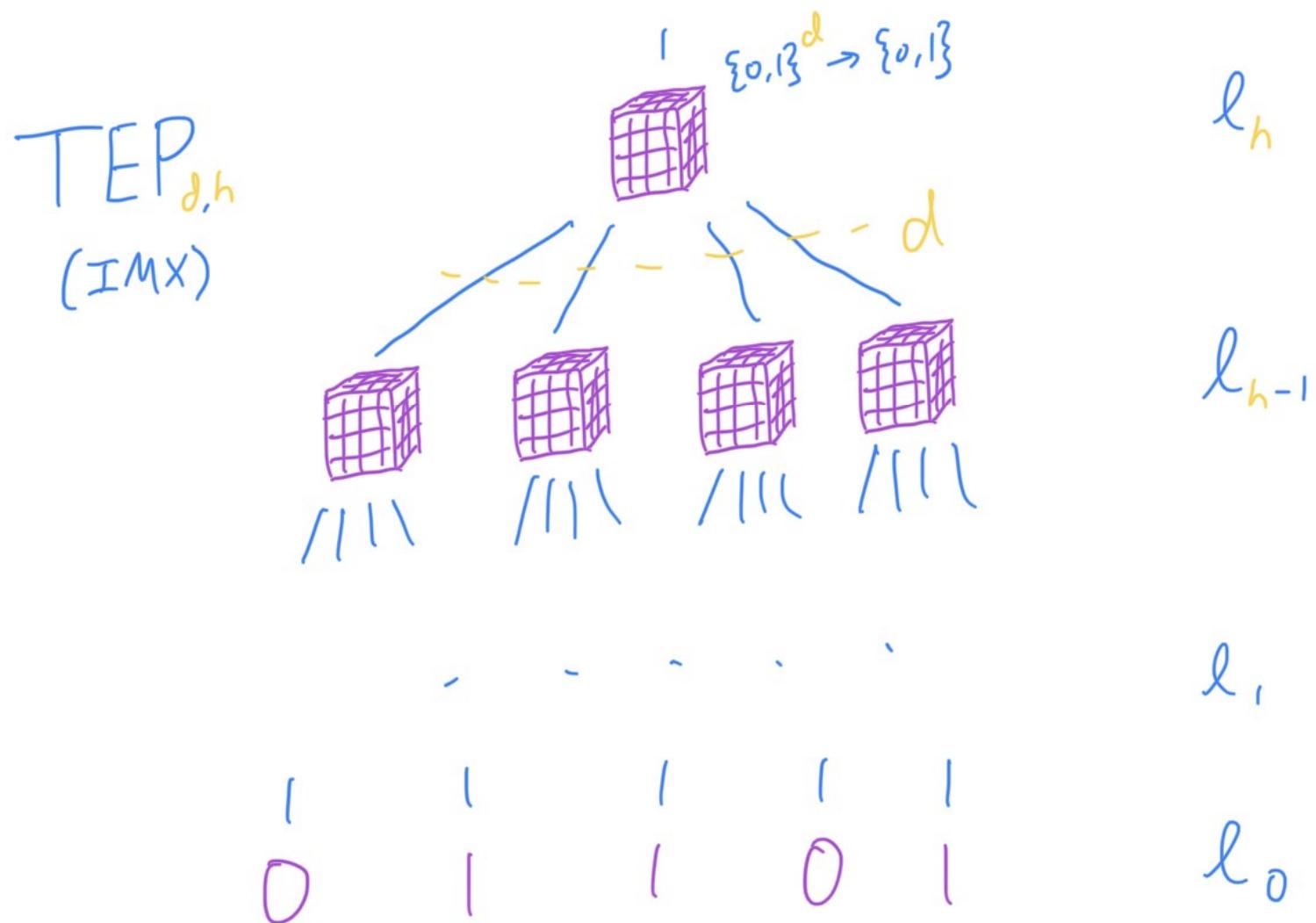
TREE EVALUATION



TREE EVALUATION



TREE EVALUATION



TREE EVALUATION

$\text{TEP}_{k,d,h} \in P$
(or even NC^2)

TREE EVALUATION

$\text{TEP}_{k,d,h} \in P$
(or even NC^2)

$NC^1 \leq \text{TEP}_{2,2,\log n}$

TREE EVALUATION

$$NC^1 \subseteq L \subseteq NL \subseteq NC^2 \subseteq P$$

TEP

HARDNESS OF TEP

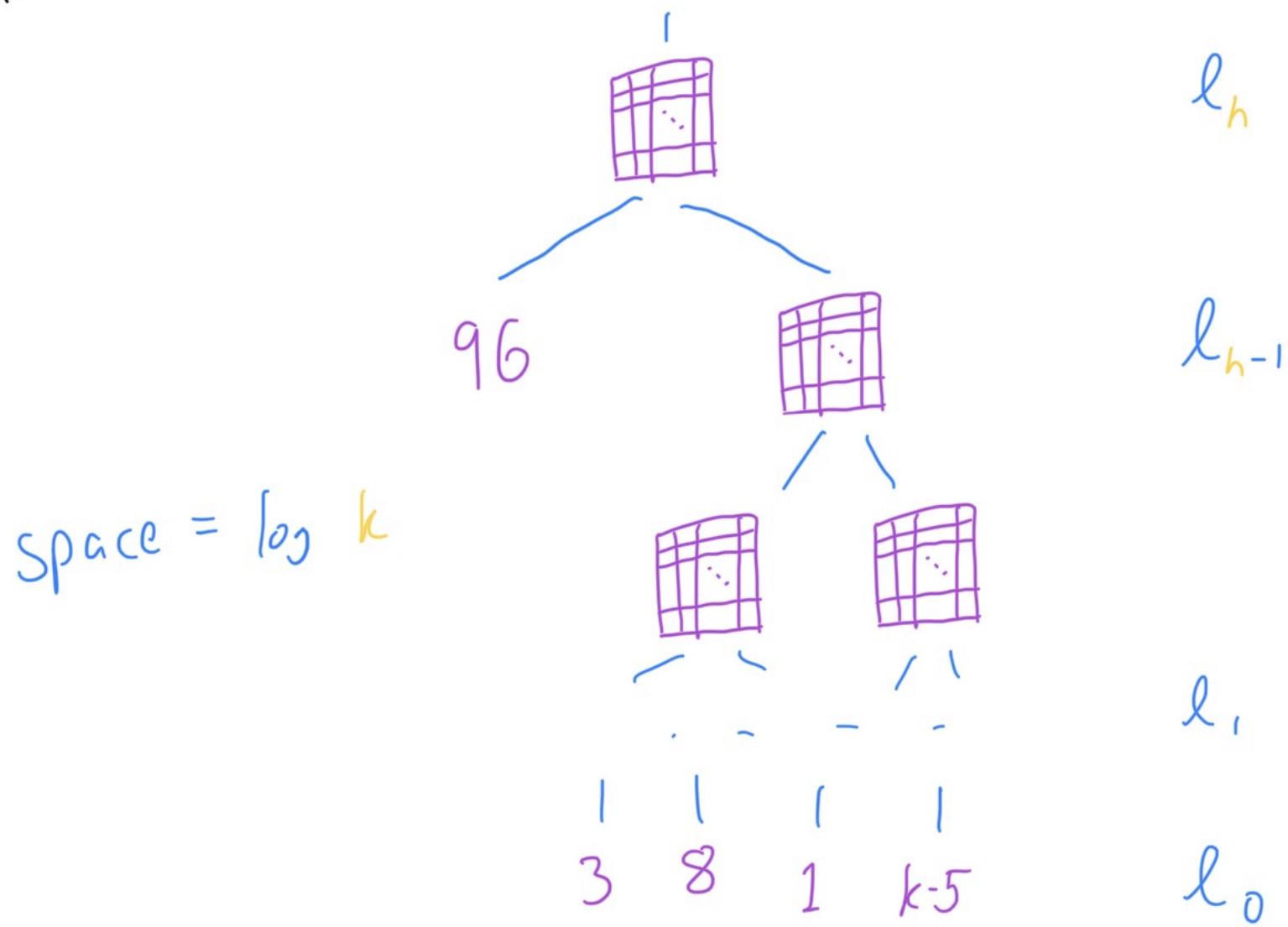
CONJECTURE [KRW'95]: $\text{TEP}_{d,h} \notin \text{NC}^1$

HARDNESS OF TEP

CONJECTURE [KRW'95]: $\text{TEP}_{d,h} \notin \text{NC}^1$

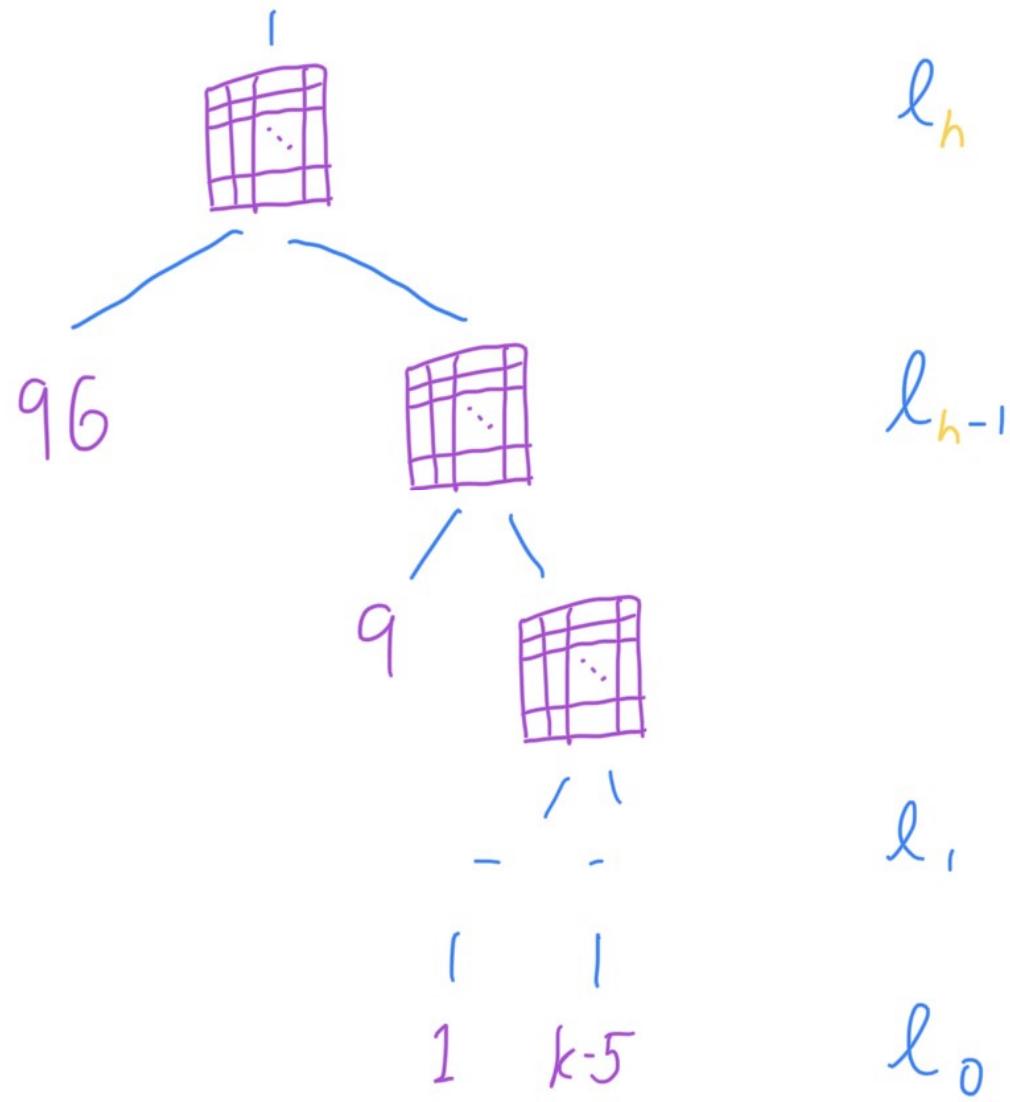
CONJECTURE [CMWBS'12]: $\text{TEP}_{k,h} \notin \text{L}$

HARDNESS OF TEP

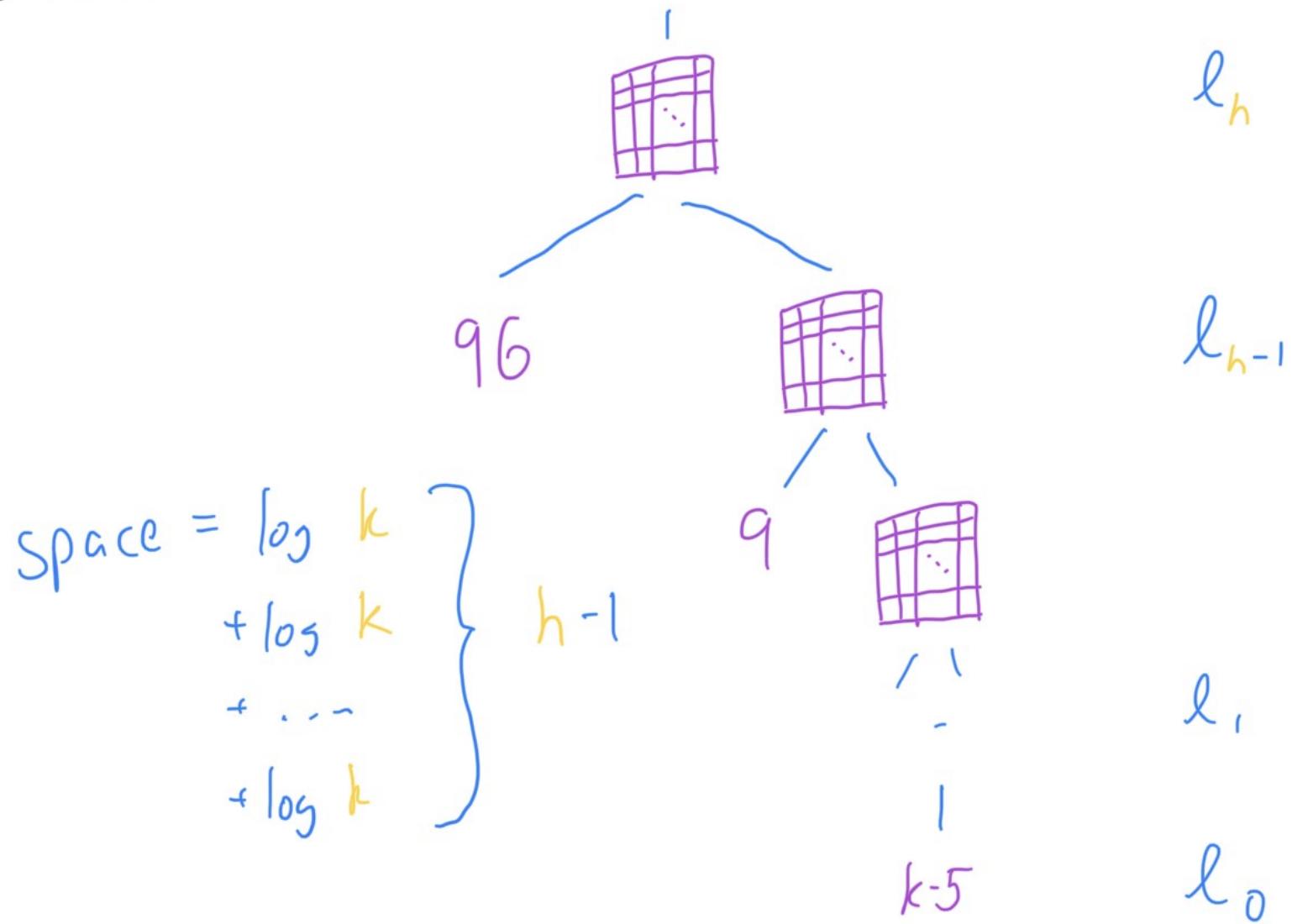


HARDNESS OF TEP

$$\text{Space} = \log k + \log k$$



HARDNESS OF TEP



HARDNESS OF TEP

CONJECTURE [CMWBS'12]: $\text{TEP}_{k,h}$ requires

space $\Omega(h \log k) = \Omega(\log^2 n)$

$$(|\text{TEP}_{k,h}| = 2^h \text{poly } k)$$

HARDNESS OF TEP

CONJECTURE [CMWBS'12]: $\text{TEP}_{k,h}$ requires

space $\Omega(h \log k) = \Omega(\log^2 n)$

thrifty ✓
read-once ✓

$$(|\text{TEP}_{k,h}| = 2^h \text{poly } k)$$

HARDNESS OF TEP

CONJECTURE [CMWBS'12]: $\text{TEP}_{k,h}$ requires

space $\Omega(h \log k) = \Omega(\log^2 n)$

non-deterministic

thrifty ✓
read-once ✓

$$(|\text{TEP}_{k,h}| = 2^h \text{poly } k)$$

HARDNESS OF TEP

CONJECTURE [KRW'95]: $\text{TEP}_{d,h}$ requires

depth $\Omega(dh) = \Omega(\log^2 n / \log \log n)$

$$(|\text{TEP}_{d,h}| = d^h 2^d)$$

EASINESS OF TEP?

BARRINGTON'S THEOREM: NC^1 can be
computed with permutation BPs
of $\text{poly}(n)$ length and width 5.

EASINESS OF TEP?

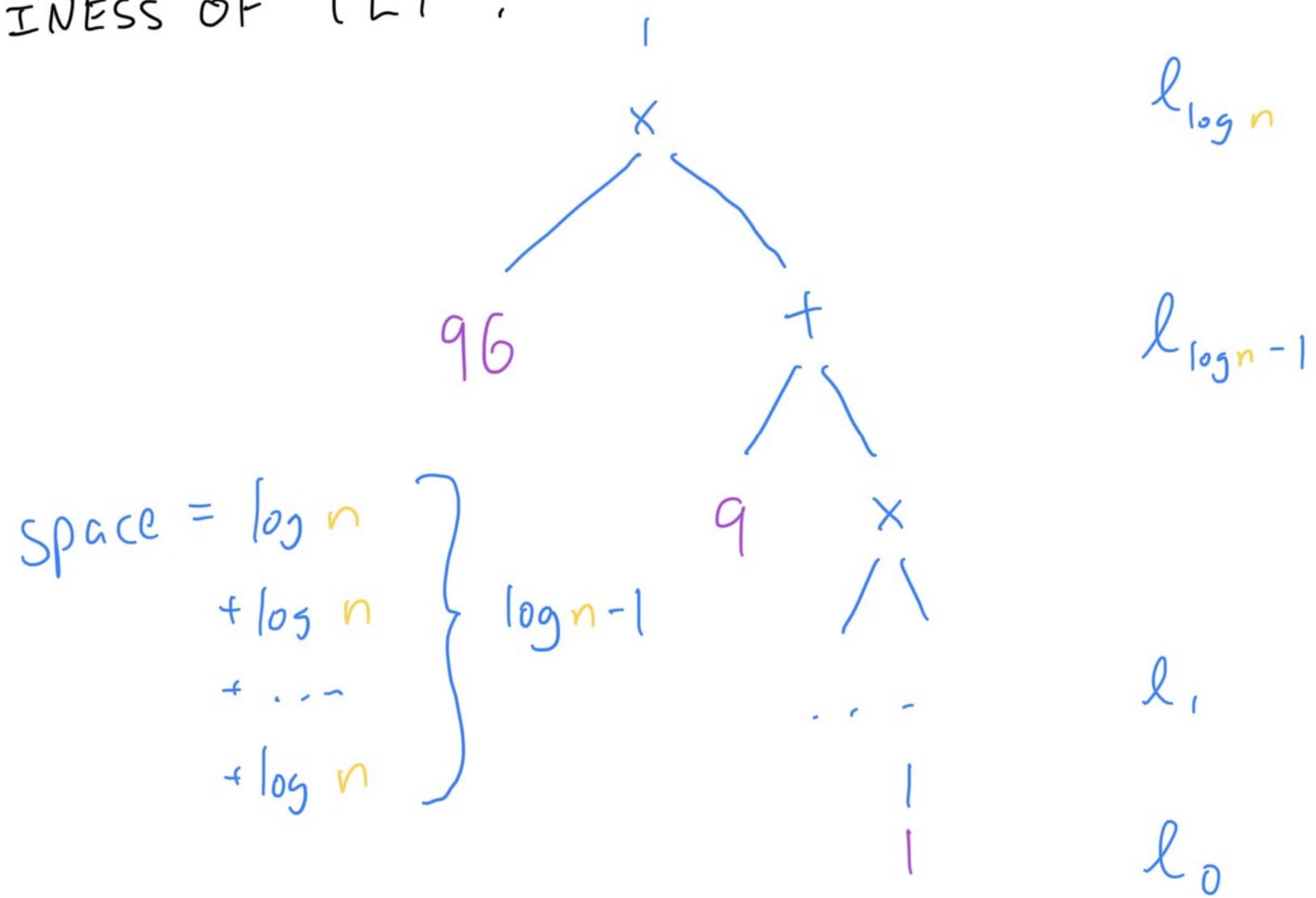
THEOREM [BC'89]: $\#NC^1 \subseteq L$.

EASINESS OF TEP?

$$\#NC^1 \quad \left(\begin{array}{c} + \\ \diagup \quad \diagdown \\ x \quad x \\ \vdots \quad \vdots \quad \vdots \\ o \quad i \quad \dots \quad o \end{array} \right) \}^{\log n}$$

$$\#NC^1 \in NC^1 \subseteq L \subseteq NL \subseteq NC^2 \subseteq P$$

EASINESS OF TEP?



BEN-OR & CLEVE

PROOF [BC'89]:

two uses of space:

1) storage

2) computation

BEN-OR & CLEVE

PROOF [BC'89]:

two uses of space:

1) storage

2) computation

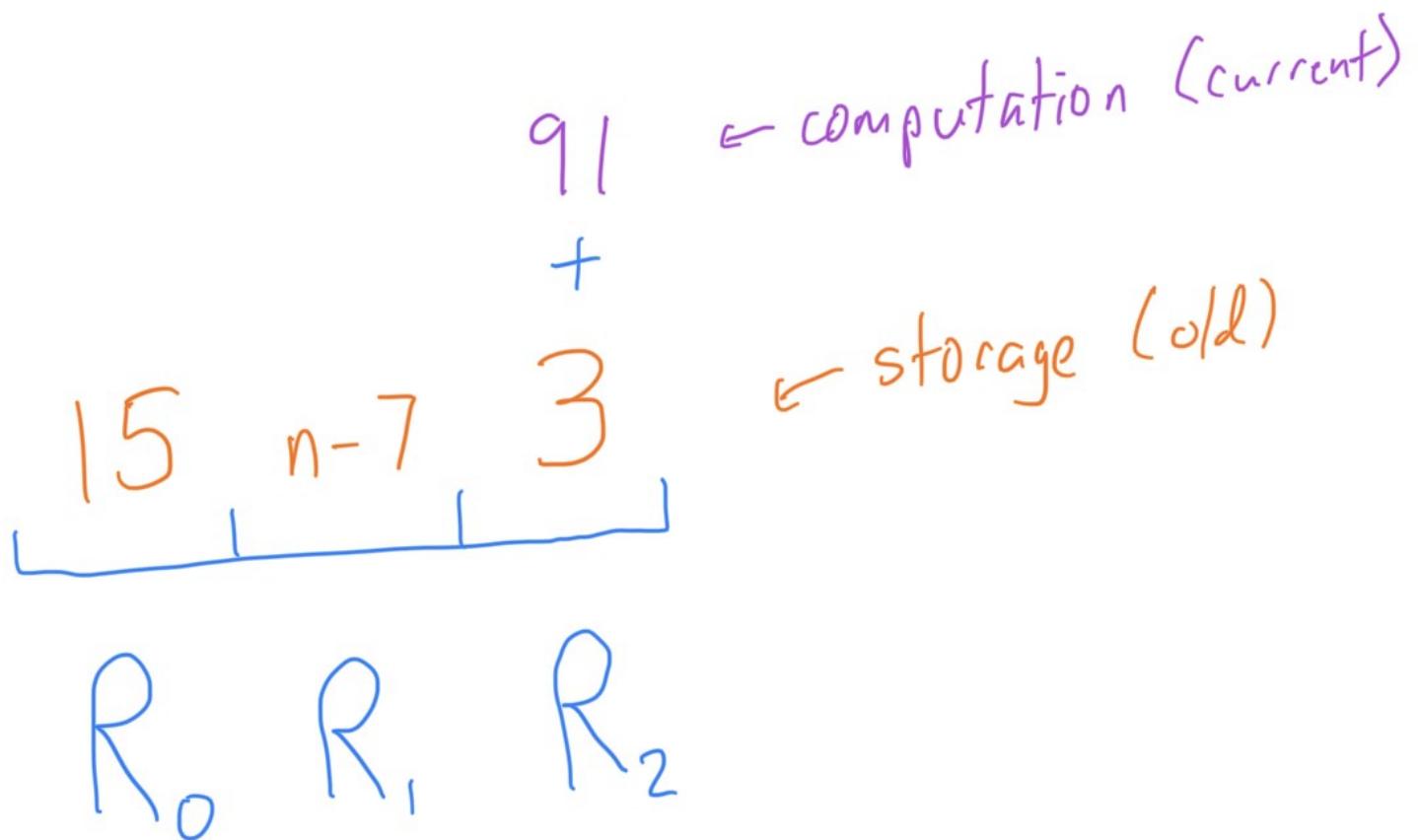
⟩ BOTH AT ONCE?

BEN-OR & CLEVE



$R_0 \quad R_1 \quad R_2$

BEN-OR & CLEVE



BEN-OR & CLEVE

$$\begin{array}{r} q_1 \times 2 \\ + \quad \quad \quad 2 \\ + \quad \quad \quad q_1 \\ \hline 15 \quad n-7 \quad 3 \end{array}$$

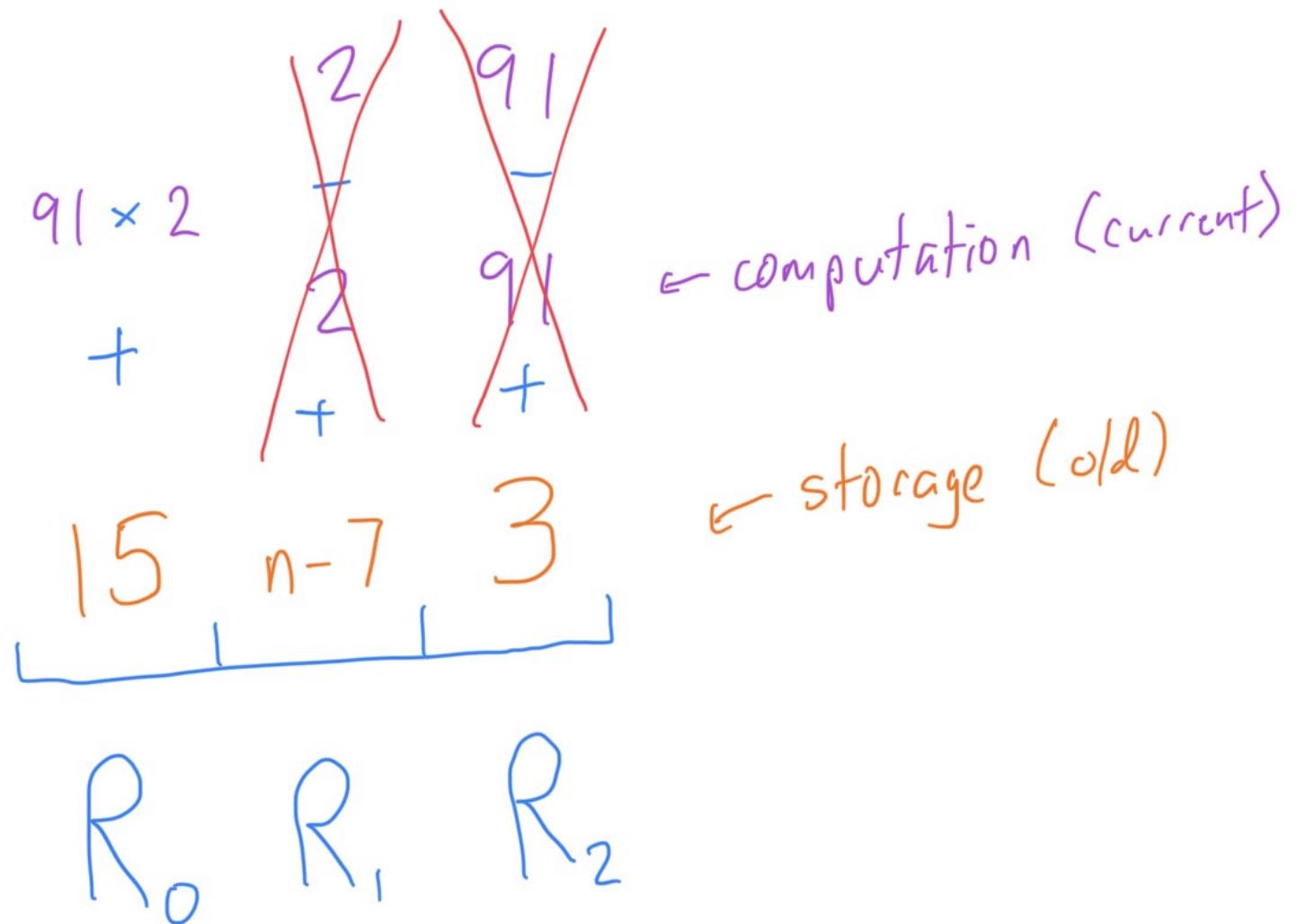
computation (current)

storage (old)

$R_0 \quad R_1 \quad R_2$

A handwritten diagram showing a vertical column of numbers: $q_1 \times 2$, 2 , and q_1 . Below this is a horizontal line with three digits: 15 , $n-7$, and 3 . Brackets above the first two digits indicate they are part of the current computation. Brackets above the last two digits indicate they are old storage values. Below the line, the labels R_0 , R_1 , and R_2 are written under the first, second, and third digits respectively.

BEN-OR & CLEVE



BEN-OR & CLEVE

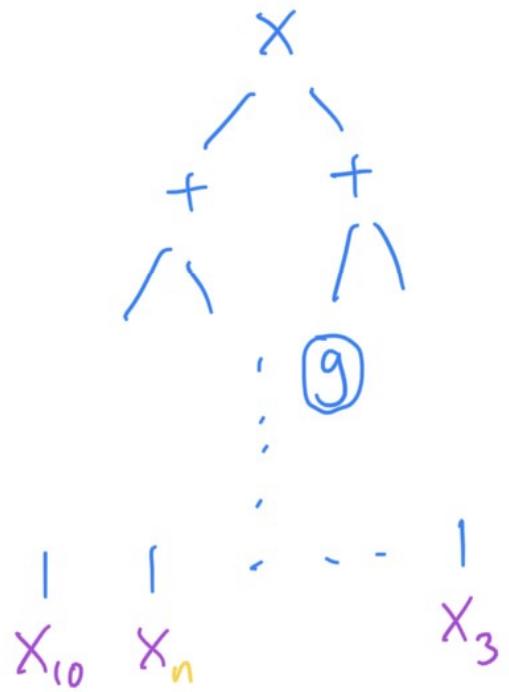
LEMMA: $\forall g \in \mathcal{C}, \exists P_g$ s.t.

$$P_g : R_0 \leftarrow R_0 + v_g$$

$$R_1 \leftarrow R_1$$

$$R_2 \leftarrow R_2$$

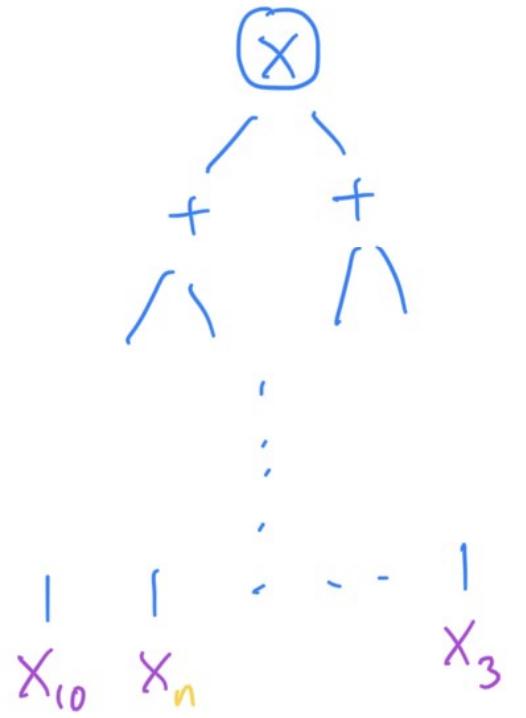
BEN-OR & CLEVE



P_g

V_g
+
 T_0 T_1 T_2
 R_0 R_1 R_2

BEN-OR & CLEVE

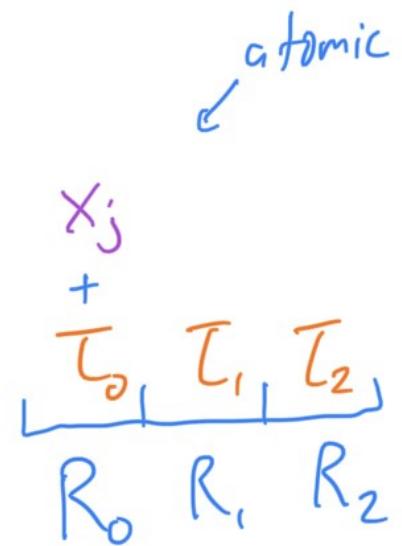
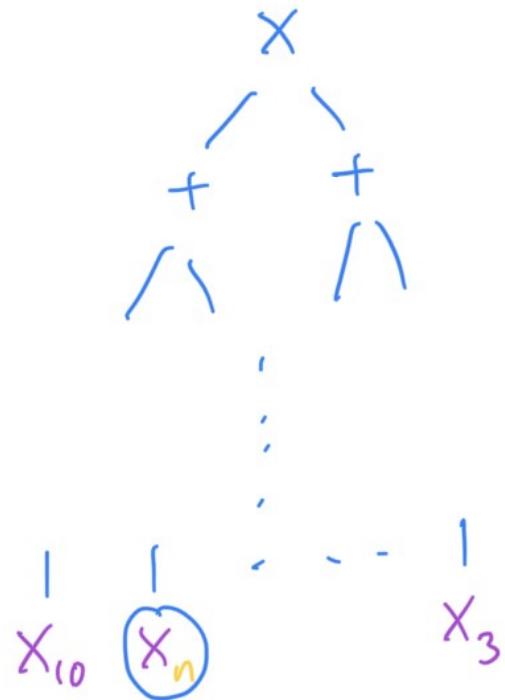


P_{out}

v_{out} ✓
+
 $\underbrace{O, O, O}_{R_0, R_1, R_2}$

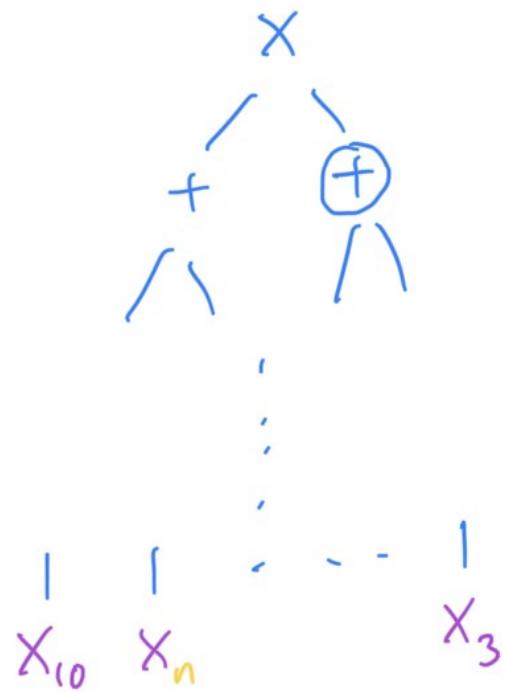
BEN-OR & CLEVE

base case: $s = x_j$



BEN-OR & CLEVE

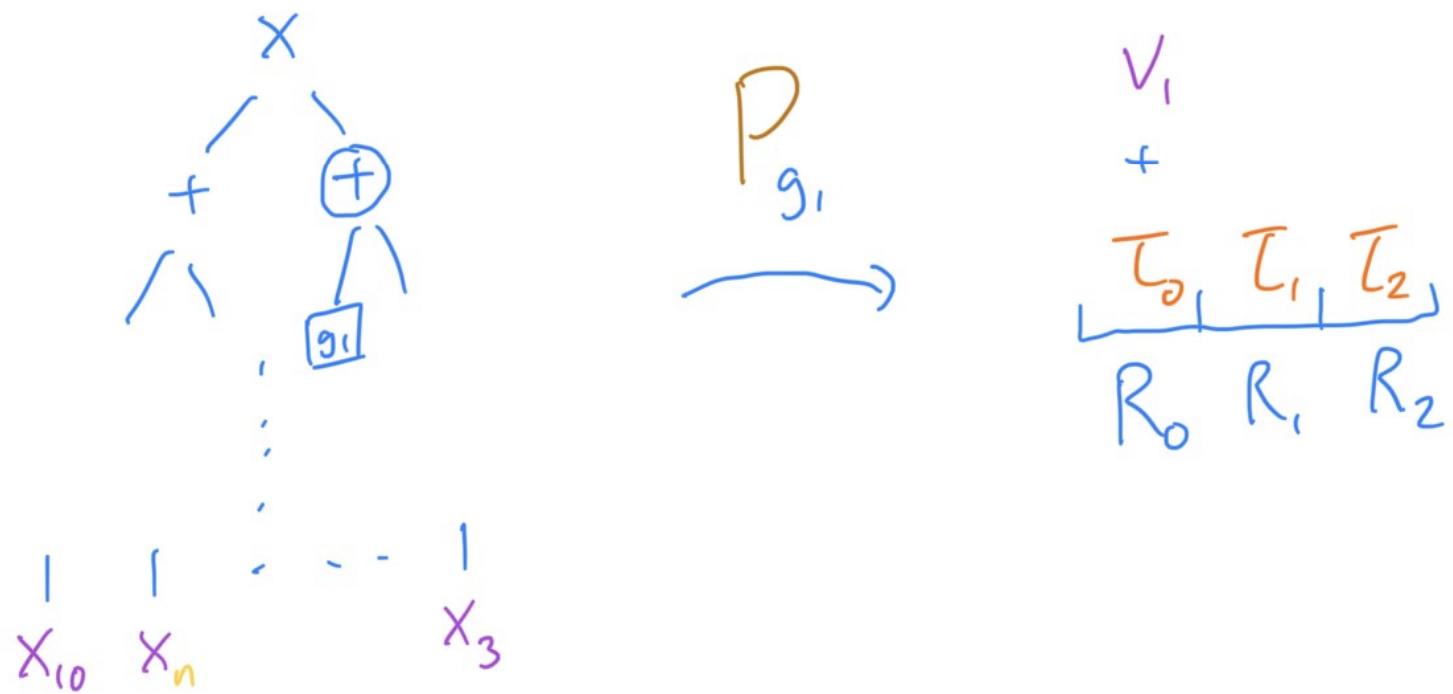
case 1 : $g = g_1 + g_2$



$$\underbrace{I_0 \quad I_1 \quad I_2}_{R_0 \quad R_1 \quad R_2}$$

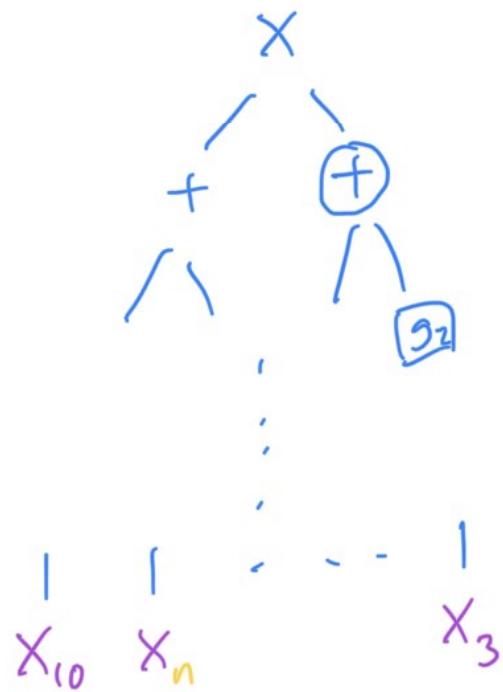
BEN-OR & CLEVE

case 1 : $g = g_1 + g_2$



BEN-OR & CLEVE

case 1 : $g = g_1 + g_2$

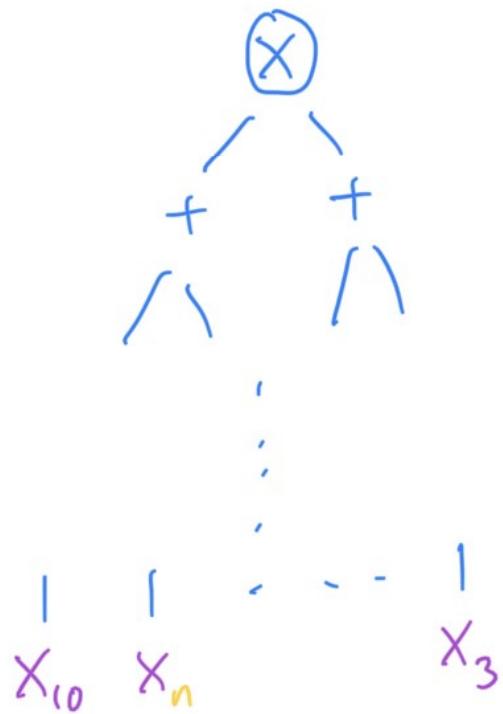


$$P_{g_2} \rightarrow$$

v_2 ✓
+
 v_1
+
 $T_0 \quad T_1 \quad T_2$
 $R_0 \quad R_1 \quad R_2$

BEN-OR & CLEVE

case 2 : $g = g_1 \times g_2$



→

$v_1 v_2$ ✓
+
 $\overbrace{I_0 \quad I_1 \quad I_2}$
 $R_0 \quad R_1 \quad R_2$

BEN-OR & CLEVE

$$\text{space} = 3 \log n + O(\log n)$$

$R_0 \ R_1 \ R_2 \qquad \text{MISC}$

+ log runtime

BEN-OR & CLEVE

$$\text{space} = 3 \log n + O(\log n)$$

$R_0 \ R_1 \ R_2 \qquad \text{MISC}$

+ log runtime

$P_g \leftarrow$ 4 calls to $P_{g'}$ s + 4 other instructions

BEN-OR & CLEVE

$$\text{space} = 3 \log n + O(\log n)$$

$R_0 \quad R_1 \quad R_2 \quad \text{MISC}$

+ log runtime

$P_g \leftarrow 4$ calls to $P_{g'}$ s + 4 other instructions

$R(h) \leq 4 \cdot R(h-1) + 4 \rightarrow 4^{\log n} \text{ poly } n \text{ instructions} \quad \square$

$\text{SPACE}(f, z) = \text{SPACE}(f) + |z|?$

$\text{SPACE}(f, z) = \text{SPACE}(f) + |z|?$

[CMWBS'12] : YES for most f (conjecture)

[BC'89] : NO for $f = +, \times$

CATALYTIC COMPUTING

[BCKLS'14]: let's model it

CATALYTIC COMPUTING

[BCKLS'14]: let's model it

input



work



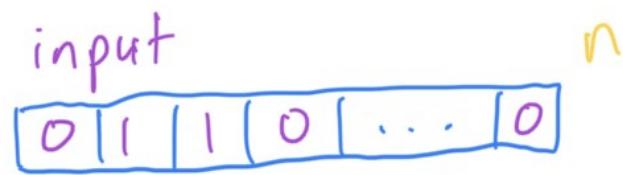
catalytic



must reset
at the end!

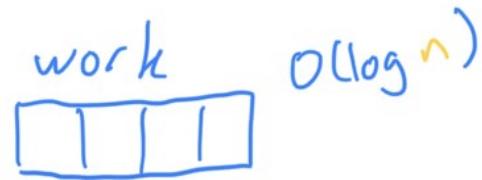
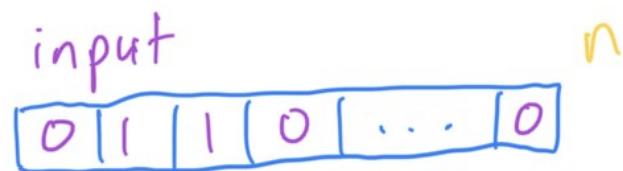
CATALYTIC COMPUTING

CL



CATALYTIC COMPUTING

CL



Q : what can CL do that L can't?

CATALYTIC COMPUTING

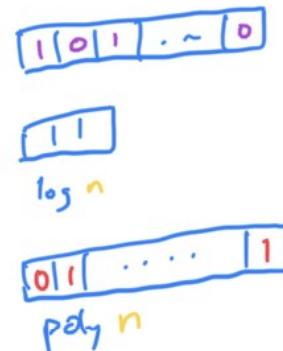
TC' is

$$NC^1 \subseteq L \subseteq NL = NC^2 \subseteq P$$

CATALYTIC COMPUTING

A tree diagram illustrating the derivation of the prefix 'MAJ' from the word 'log'. The root node is 'TC'. It branches into three 'MAJ' nodes. Each 'MAJ' node further branches into three smaller 'MAJ' nodes, which then branch into individual letters: 'l', 'o', and 'g'. Brackets on the right side group the three main 'MAJ' nodes under the label 'log'.

CL



$$TC'_{in} = CL$$

$$NC^1 \subseteq L \subseteq NL = NC^2 \subseteq P$$

CATALYTIC COMPUTING

LEMMA: $\forall g \in \mathcal{C}, \exists P_g$ s.t.

$$P_g : R_0 \leftarrow R_0 + v_g$$

$$R_i \leftarrow R_i \quad \forall i \neq 0$$

gateset: $\text{MAJ}(x) = \begin{cases} 1 & \text{if } \sum_{i=1}^m x_i \geq \frac{m}{2} \\ 0 & \text{o.w.} \end{cases}$

CATALYTIC COMPUTING

$$\text{MAJ}(x) \equiv \sum_{k=\frac{m}{2}}^m \left[1 - \left(\sum_{i=1}^m x_i - k \right)^{p-1} \right] \bmod p$$

CATALYTIC COMPUTING

$$\text{MAJ}(x) \equiv \sum_{k=\frac{m}{2}}^m \left[1 - \left(\sum_{i=1}^m x_i - k \right)^{p-1} \right] \bmod p$$

P_{Σ} $P_{\wedge p-1}$

The diagram illustrates the components of the MAJ function. A bracket under the summation sign $\sum_{k=\frac{m}{2}}^m$ points to the symbol P_{Σ} . Another bracket under the power term $(\sum_{i=1}^m x_i - k)^{p-1}$ points to the symbol $P_{\wedge p-1}$.

CATALYTIC COMPUTING

$$\text{MAJ}(x) \equiv \sum_{k=\frac{m}{2}}^m \left[1 - \left(\sum_{i=1}^m x_i - k \right)^{p-1} \right] \bmod p$$

P_{Σ} $P_{\wedge p-1}$

Efficiency: $\text{poly } n$ registers

$O(1)$ recursive calls

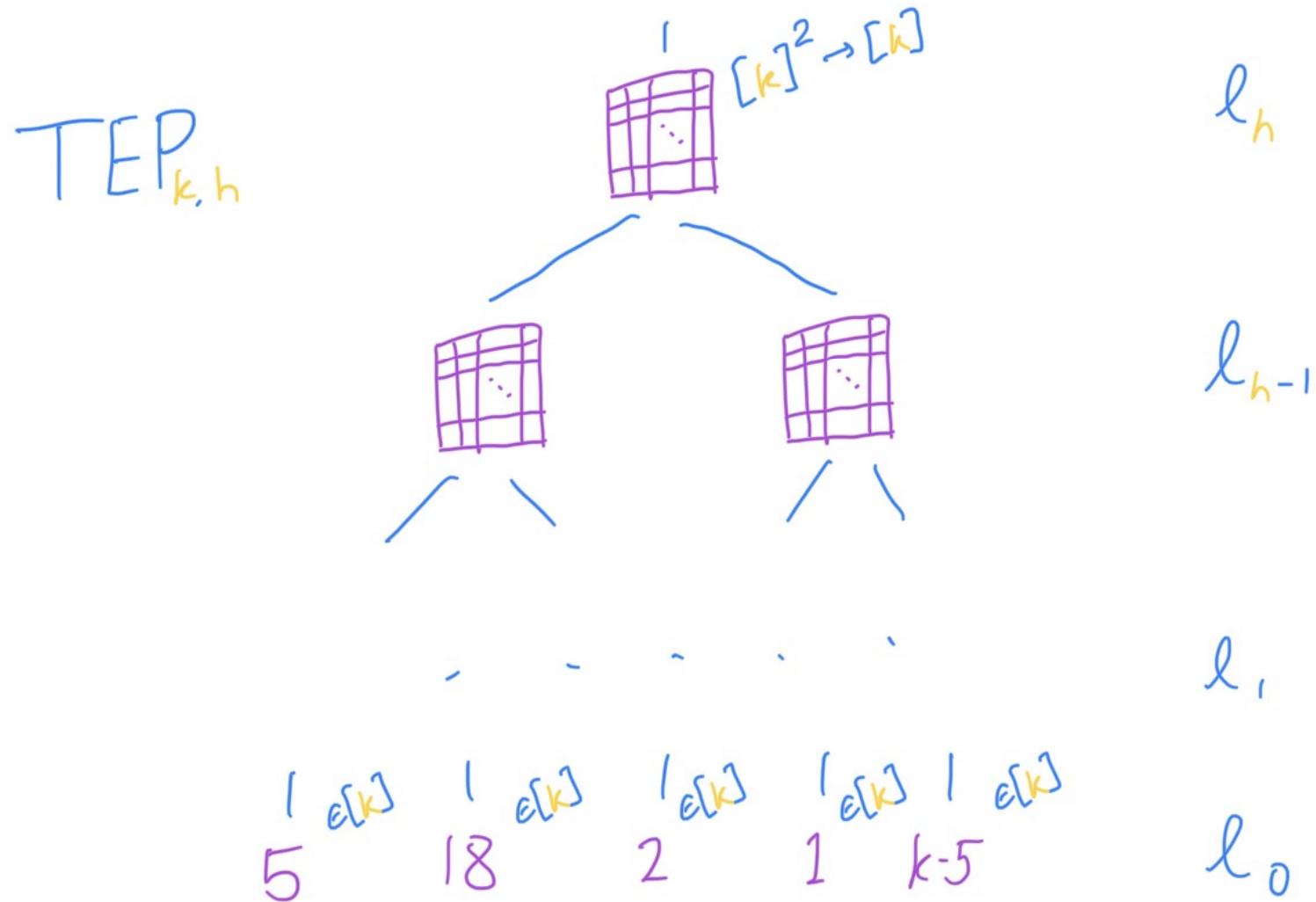
□

$\text{SPACE}(f, z) = \text{SPACE}(f) + |\underline{z}|?$

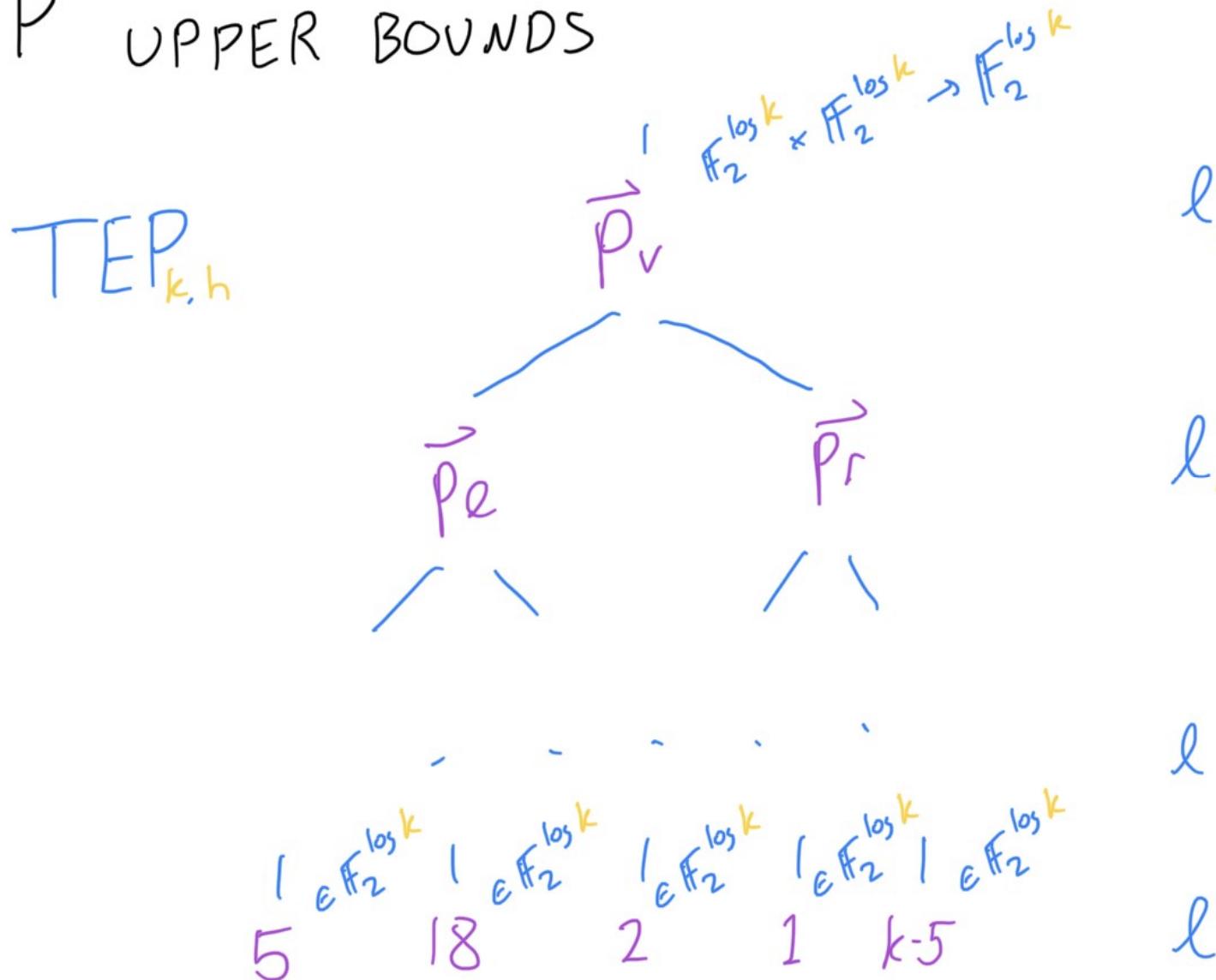
TEP UPPER BOUNDS

THEOREM [CM'20, 21]: TEP_{k,h} can be solved in space $O(h \log k / \log h)$
 $(= O(\log^2 n / \log \log n))$

TEP UPPER BOUNDS



TEP UPPER BOUNDS



TEP UPPER BOUNDS

LEMMA: $\forall g \in \mathcal{C}, \exists P_g$ s.t.

$$P_g : R_0 \leftarrow R_0 + v_g$$

$$R_i \leftarrow R_i \quad \forall i \neq 0$$

gateset: $\vec{P}(\vec{y}, \vec{z}) : \mathbb{F}_2^{\log k} \times \mathbb{F}_2^{\log k} \rightarrow \mathbb{F}_2^{\log k}$

TEP UPPER BOUNDS

$$\vec{P}(\vec{y}, \vec{z}) : \mathbb{F}_2^{\log k} \times \mathbb{F}_2^{\log k} \rightarrow \mathbb{F}_2^{\log k}$$

Efficiency: $3 \log k$ registers over \mathbb{F}_2

$$\rightarrow \text{space} = 3 \log k$$

TEP UPPER BOUNDS

$$\vec{P}(\vec{y}, \vec{z}) : \mathbb{F}_2^{\log k} \times \mathbb{F}_2^{\log k} \rightarrow \mathbb{F}_2^{\log k}$$

Efficiency: $3 \log k$ registers over \mathbb{F}_2

k^2 recursive calls

$$\rightarrow \text{space} = 3 \log k + \log (k^2)^h = h \log k \quad \square$$

TERP UPPER BOUNDS

$$\vec{P}(\vec{y}, \vec{z}) : \mathbb{F}_2^{\log k} \times \mathbb{F}_2^{\log k} \rightarrow \mathbb{F}_2^{\log k}$$

Efficiency:

$3 \log k$ registers over \mathbb{F}_2

$2^{\deg(P)}$ recursive calls

TEP UPPER BOUNDS

FEWER RECURSIVE CALLS



FASTER TEP ALGORITHMS



SMALLER SPACE TEP

$$2^{\circ(\deg(p))}$$

$$(2^{\circ(\log k)})^h$$

$$\circ(h \log k)$$

alternative
facts ahead

TED lower bounds
are a hoax

WARNING

NOT PEER REVIEWED

^
OFFICIALLY

I do my own
research

Free Speech zone

TEP UPPER BOUNDS

THEOREM [CM'23]: TEP_{k,h} can be solved in space $O(h^+ \log k) \cdot \log \log k$
 $(= O(\log n \cdot \log \log n))$

TEP UPPER BOUNDS

$\vec{p}_v(\vec{l}, \vec{r}) : O(\deg(p))$ recursive calls

$$\text{space} = \log [O(\log k)^h]$$

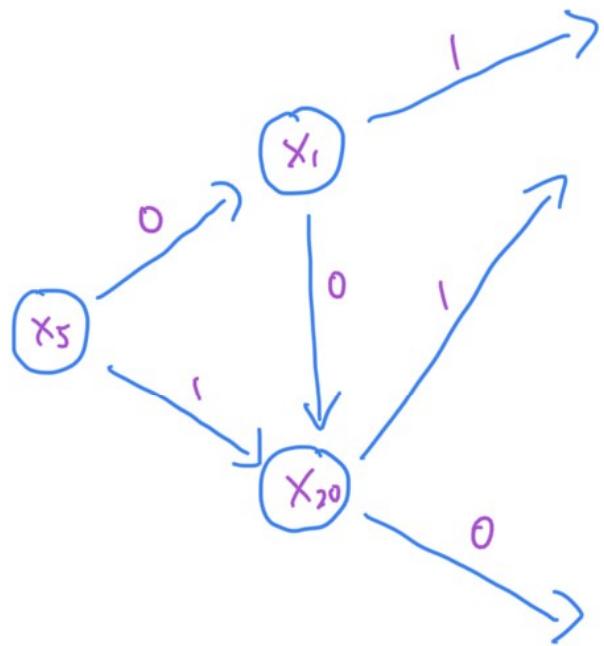
runtime

TEP UPPER BOUNDS

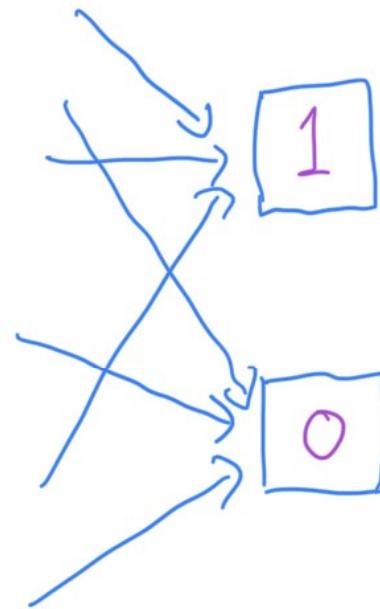
$\vec{p}_v(\vec{l}, \vec{r}) : O(\deg(p))$ recursive calls
 $3 \log k$ registers over $\mathbb{F}_{O(\deg(p))}$

$$\begin{aligned} \text{space} &= \log [O(\log k)^h] + (3 \log k) \cdot \log [O(\log k)] \\ &\quad \text{runtime} \qquad \qquad \# \text{ reg.} \qquad \qquad \text{Size per reg.} \\ &= O(h + \log k) \cdot \log \log k \end{aligned}$$

AMORTIZED BPs

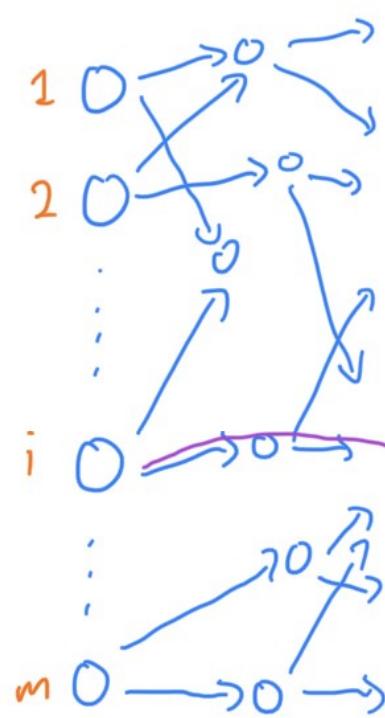


- - -



size $s \equiv$ non-uniform space $\log s$

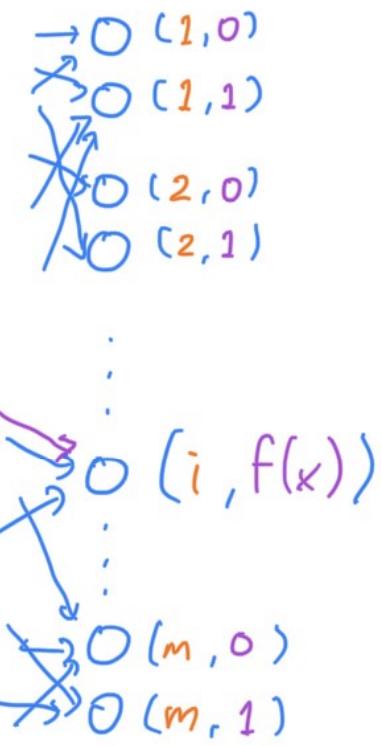
AMORTIZED BPs



size $m s$

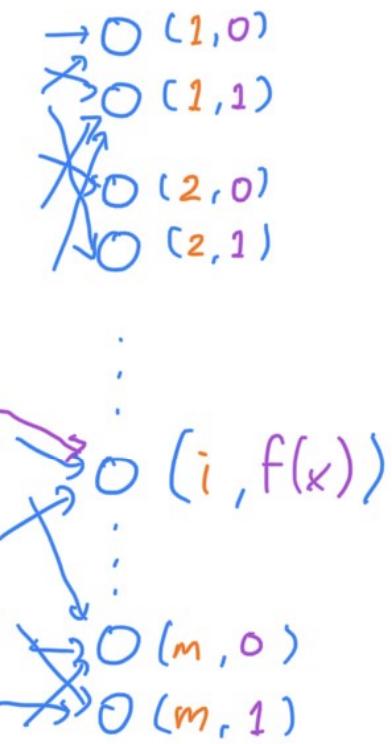
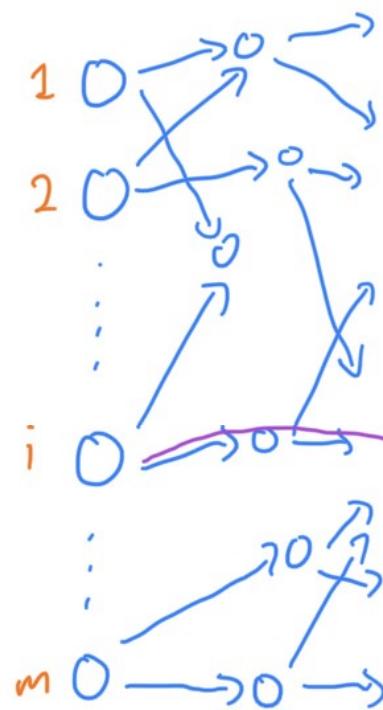
\equiv

non-uniform space $\log s$



catalytic $\log m$

AMORTIZED BPs



size $m s \equiv$ amortized m size s

AMORTIZED BPs

THEOREM [P'17] : $s = O(n)$ $m = 2^{2^n}$

AMORTIZED BPs

THEOREM [P'17] : $s = O(n)$ $m = 2^{2^n}$

THEOREM [CM'22] : $s = O_\epsilon(n)$ $m = 2^{2^{\epsilon n}}$

AMORTIZED BPs

THEOREM [P'17] : $s = O(n)$ $m = 2^{2^n}$

THEOREM [CM'22] : $s = O_\epsilon(n)$ $m = 2^{2^{\epsilon n}}$

THEOREM [CM'23] : $s = n^{2+\epsilon}$ $m = 2^{O_\epsilon(n)}$

KRW AND SPACE

CONJECTURE [KRW'95]: $\text{TEP}_{d,h} \notin \text{NC}^1$

KRW AND SPACE

CONJECTURE [KRW'95]: $\text{TEP}_{d,h} \notin \text{NC}^1$

THEOREM [CM'23]: $\text{KRW} \rightarrow \text{NC}^1 \neq L$

KRW AND SPACE

1. KRW gives a very sharp separation between complexity classes

KRW AND SPACE

1. KRW gives a very sharp separation between complexity classes
2. KRW \rightarrow quasipoly (uniform) separation between formulas and branching programs

KRW AND SPACE

1. KRW gives a very sharp separation between complexity classes
2. KRW \rightarrow quasipoly (uniform) separation between formulas and branching programs
3. formally easier to show $\text{STCONN} \notin \text{NC}^1$ than $\text{TEP} \notin \text{NC}^1$

WHAT Now?

- TEP still not in L yet!

WHAT Now?

- TEP still not in L yet!
- what is TEP complete for?

WHAT Now?

- TEP still not in L yet!
- what is TEP complete for?
- other things to do with catalytic?

SHAMELESS PLUGS

RESULTS [CM '23] : ECCC (soon !)
(longer talk to be posted on release)

SHAMELESS PLUGS

RESULTS [CM '23] : ECCC (soon !)
(longer talk to be posted on release)

SURVEY [M '23] : B. EATCS
→ ECCC
Suggestions appreciated!

THANKS !