Catalytic Computing, Tree Evaluation, & Clean Computation

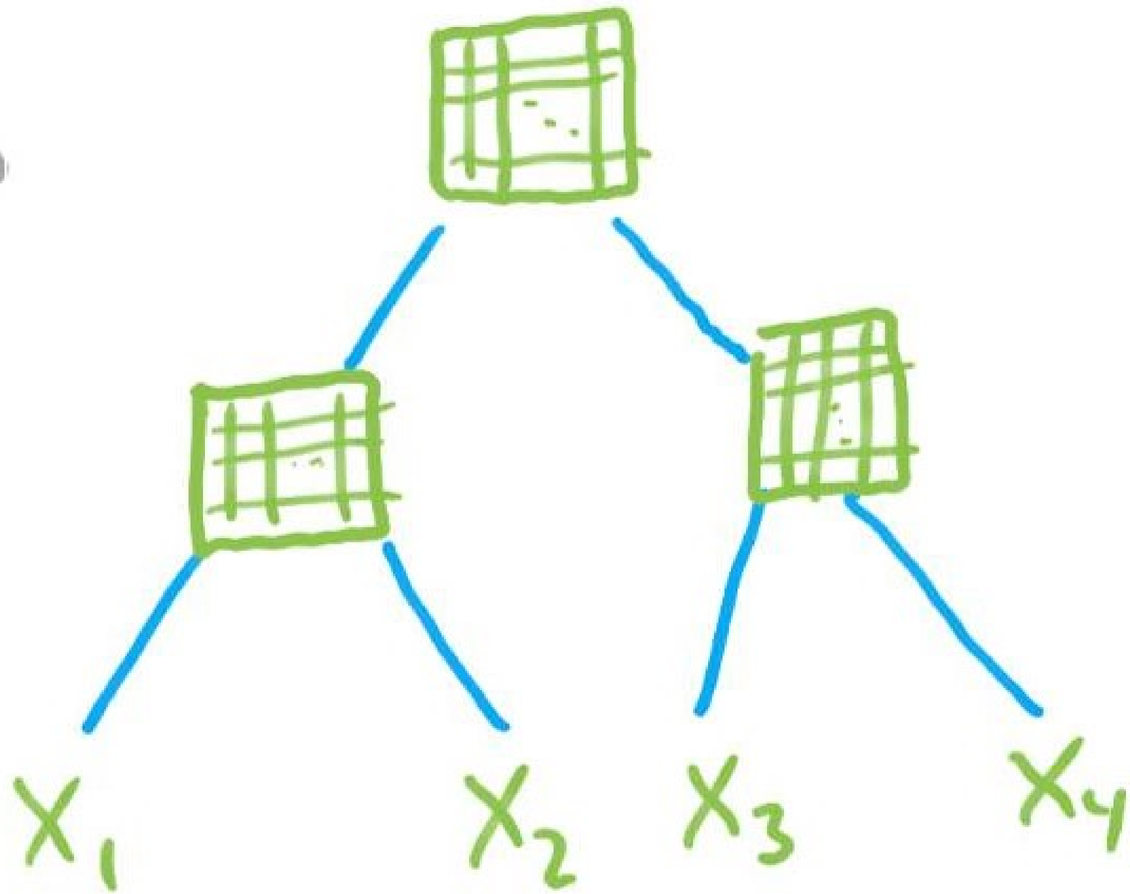
Ian Mertz

Thesis Proposal 2020.11.30

Tree Evaluation

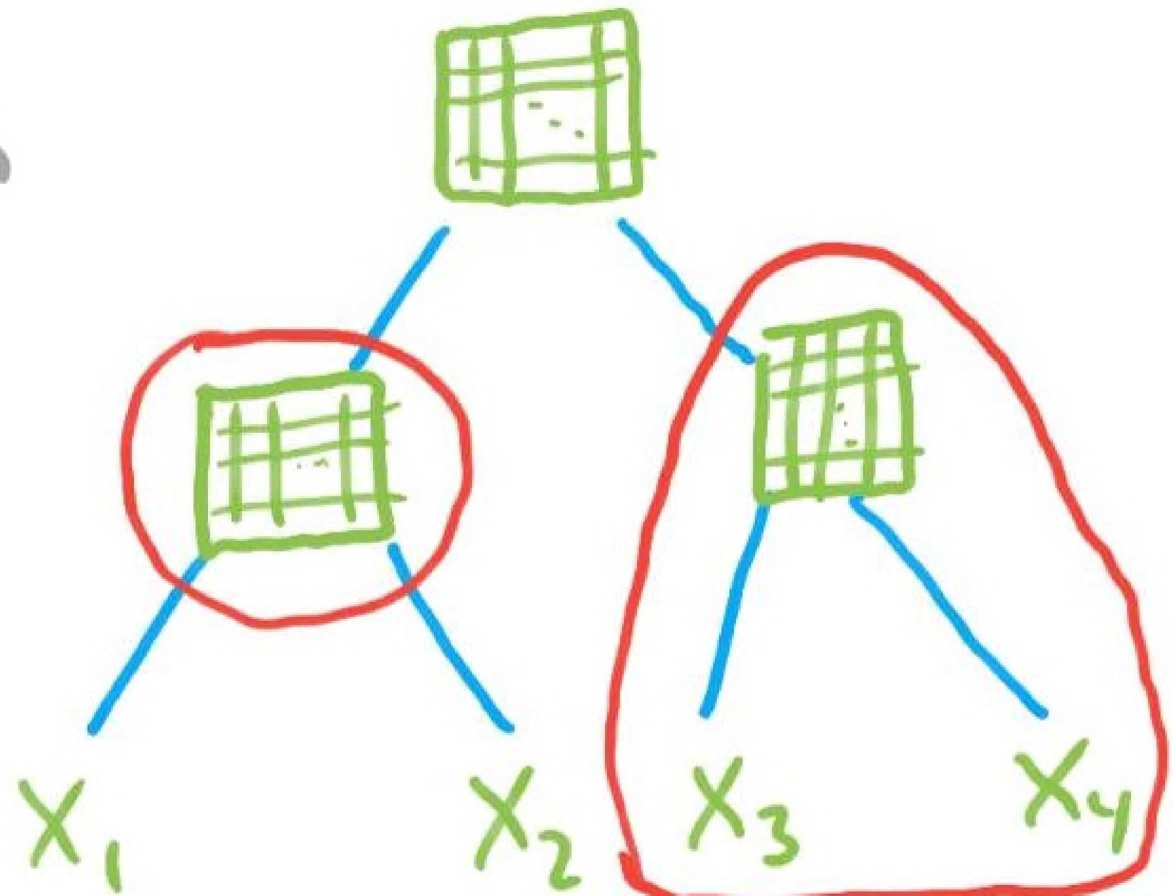
height h

domain k



Tree Evaluation

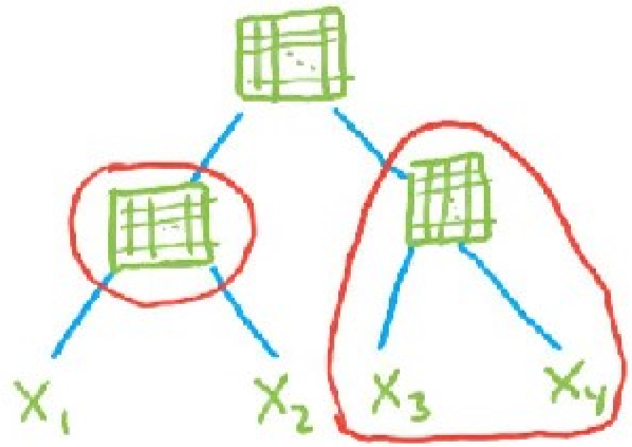
height h
domain k



Tree Evaluation

height h

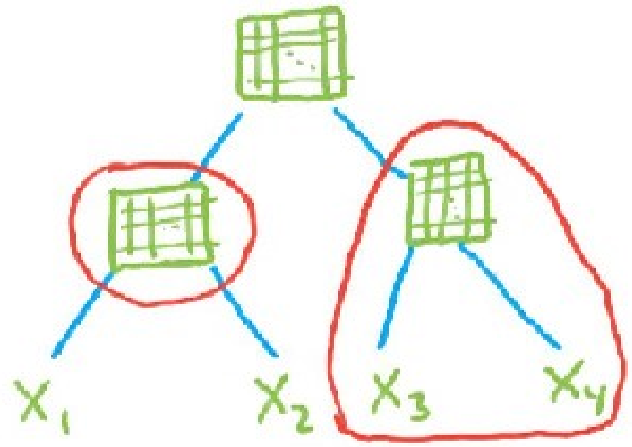
domain k



Conjecture [Cook]: $TEP_{k,h}$ requires space $\Omega(h \log k)$.

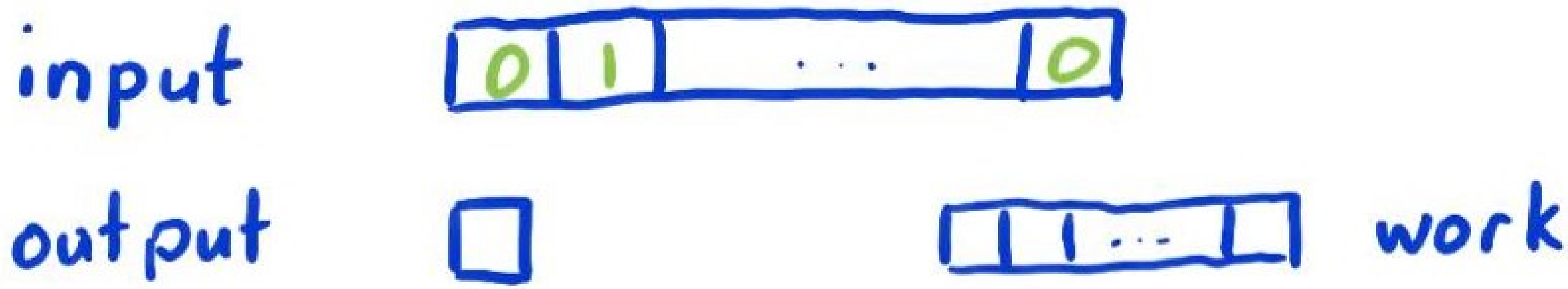
Tree Evaluation

height h
domain k

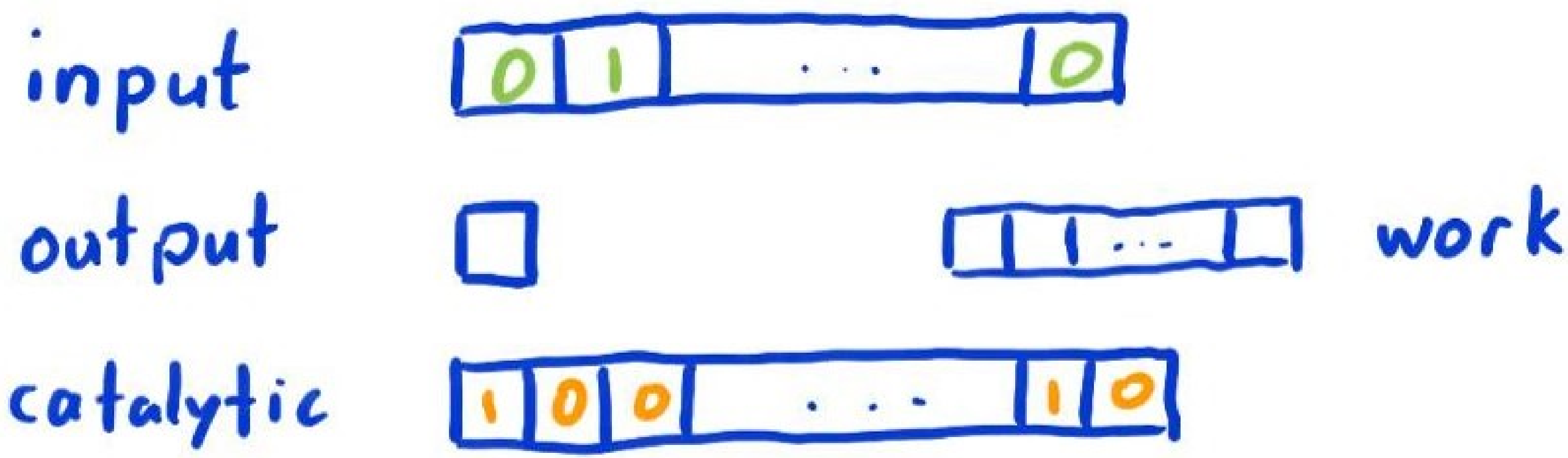


Theorem [Cook-Mertz]: $TEP_{k,h}$ can be solved in space $O(h \log k / \log h)$.

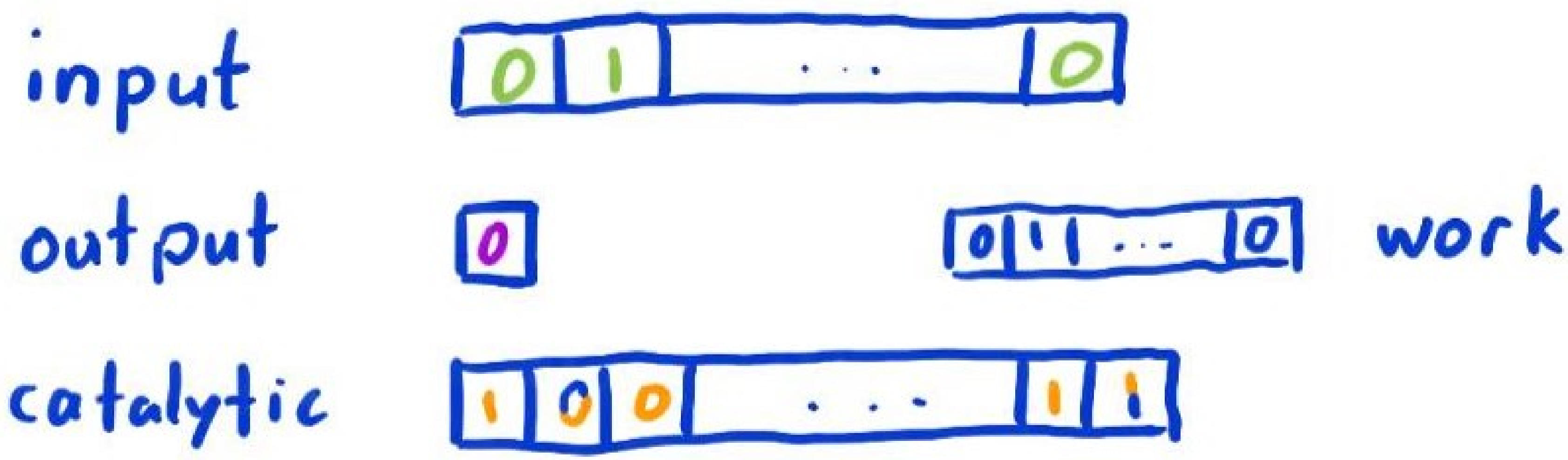
CATALYTIC COMPUTATION



CATALYTIC COMPUTATION

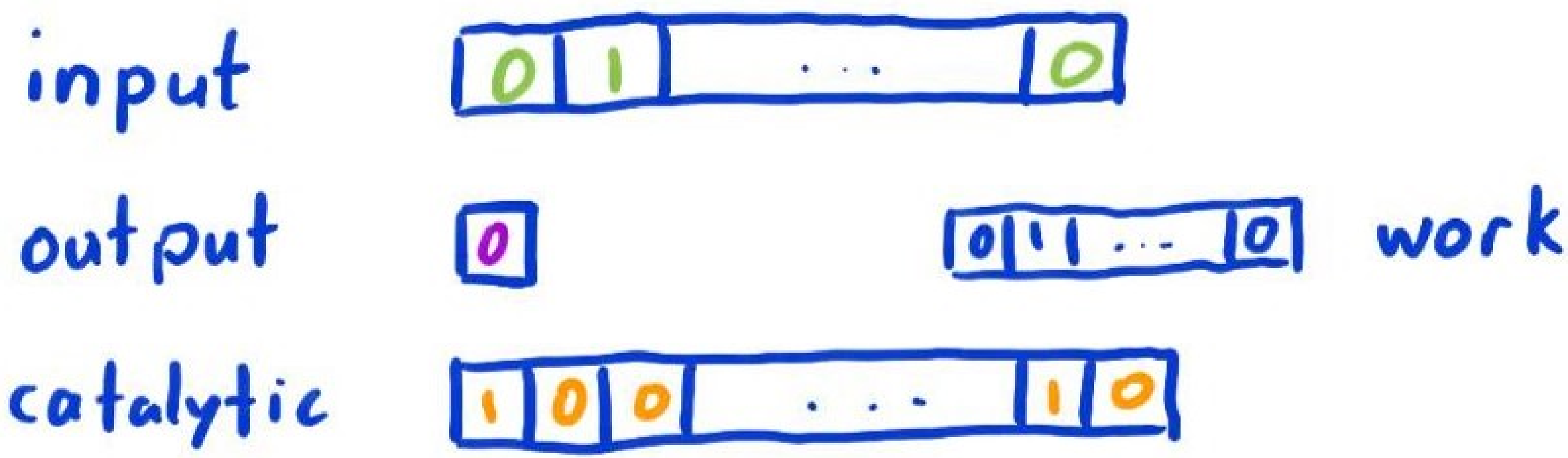


Catalytic Computation



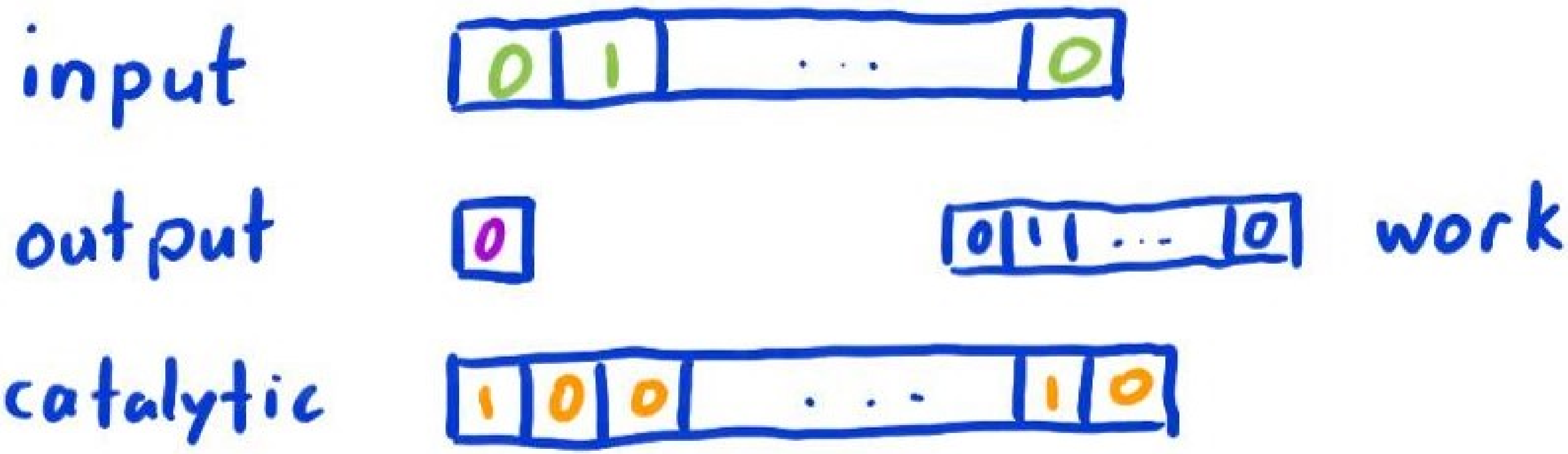
$$R_{out} = f(x_1 \ldots x_n)$$

CATALYTIC COMPUTATION



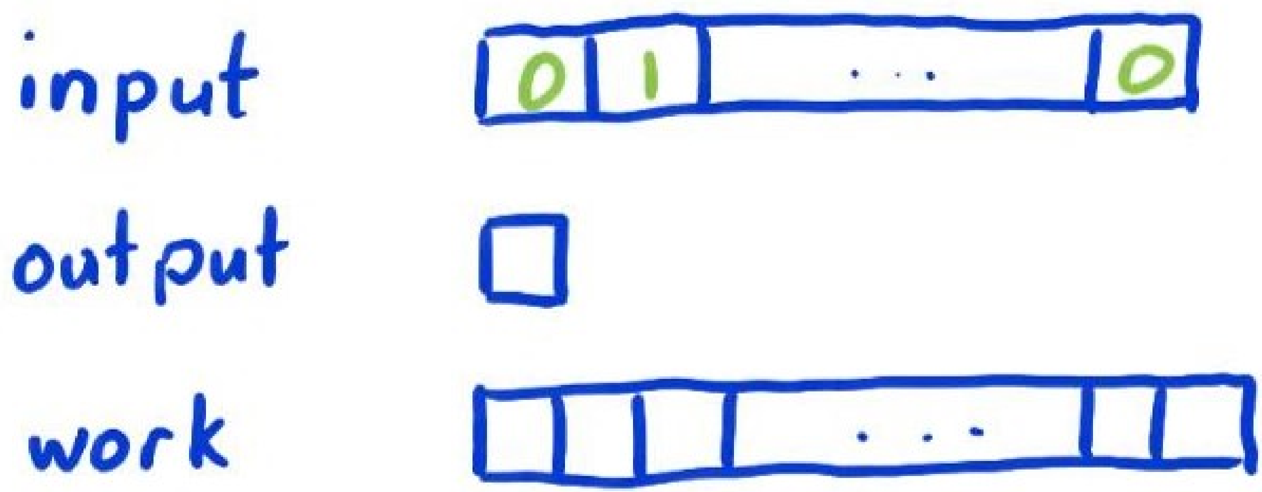
$$R_{out} = f(x_1 \ldots x_n) \qquad R_i = \tau_i$$

Catalytic Computation

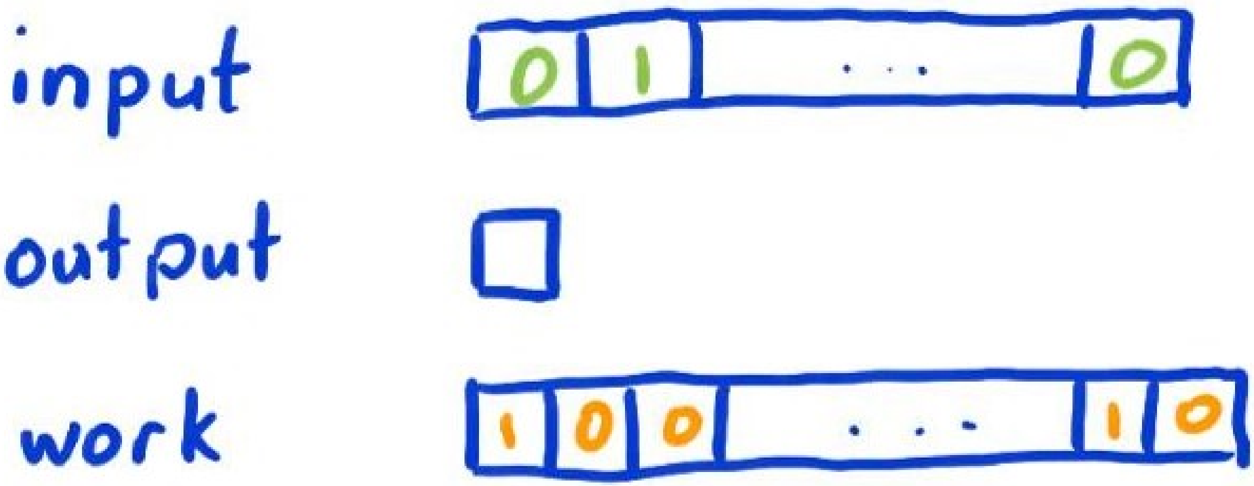


Theorem [BCKLS'14]: $\mathsf{CL} \supseteq \mathsf{TC}^1$

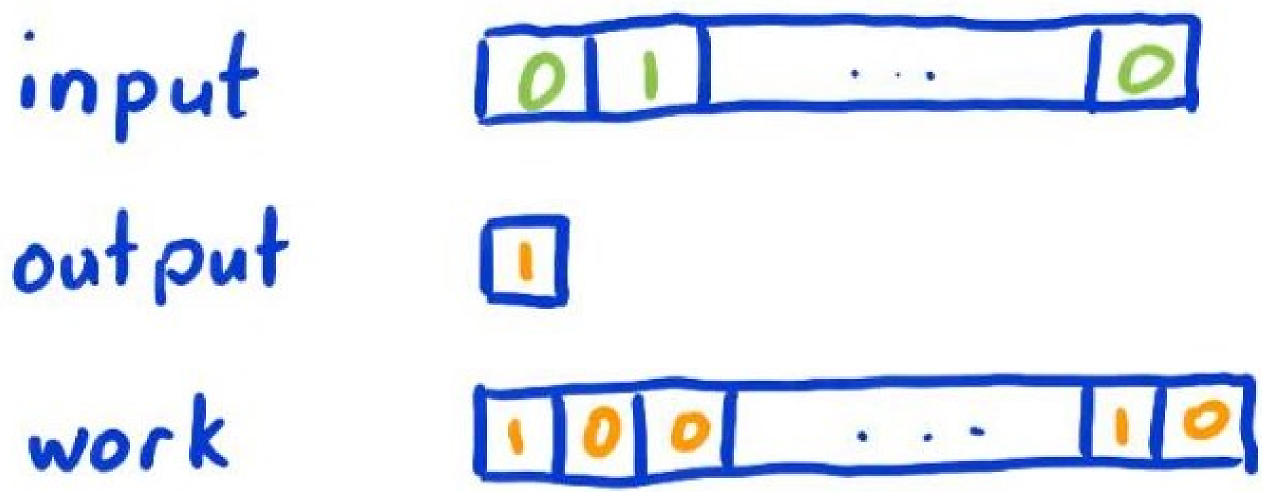
Clean Computation



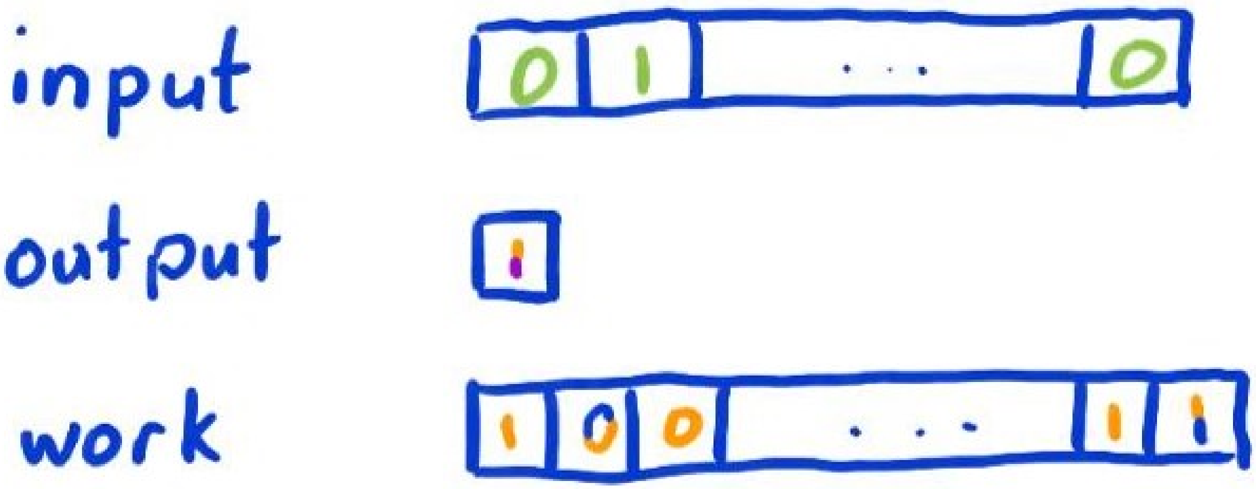
CLEAN COMPUTATION



Clean Computation

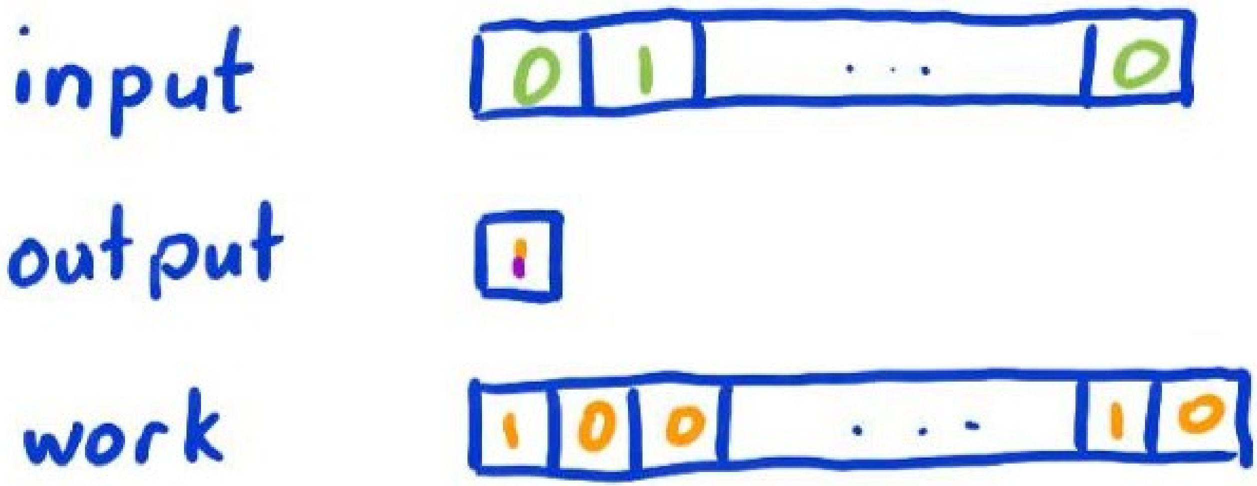


Clean Computation



$$R_{out} = f(x_1 \ldots x_n) + \tau_{out}$$

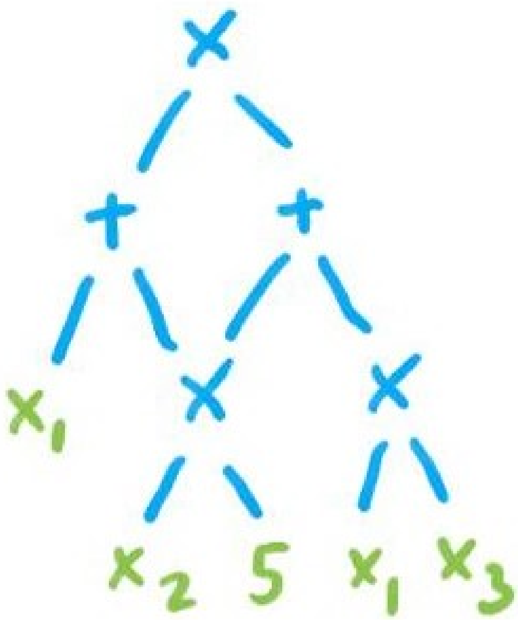
Clean Computation

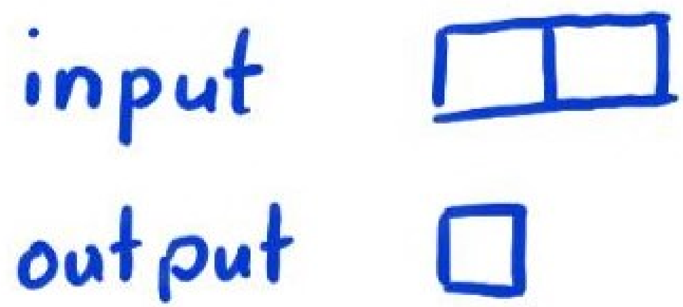


$$R_i = \tau_i$$

$$R_{out} = f(x_1 \ldots x_n) + \tau_{out}$$

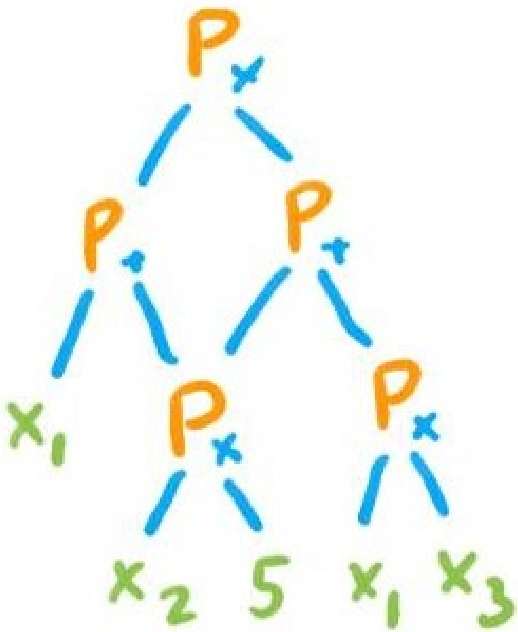
Clean Computation

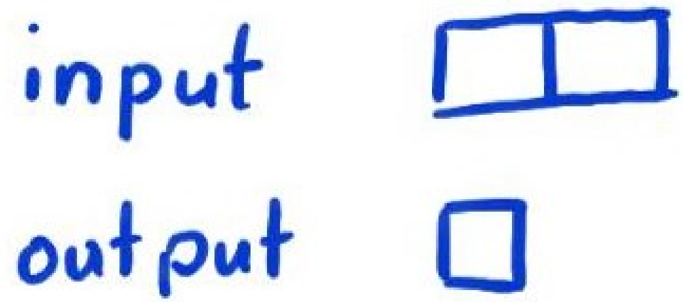




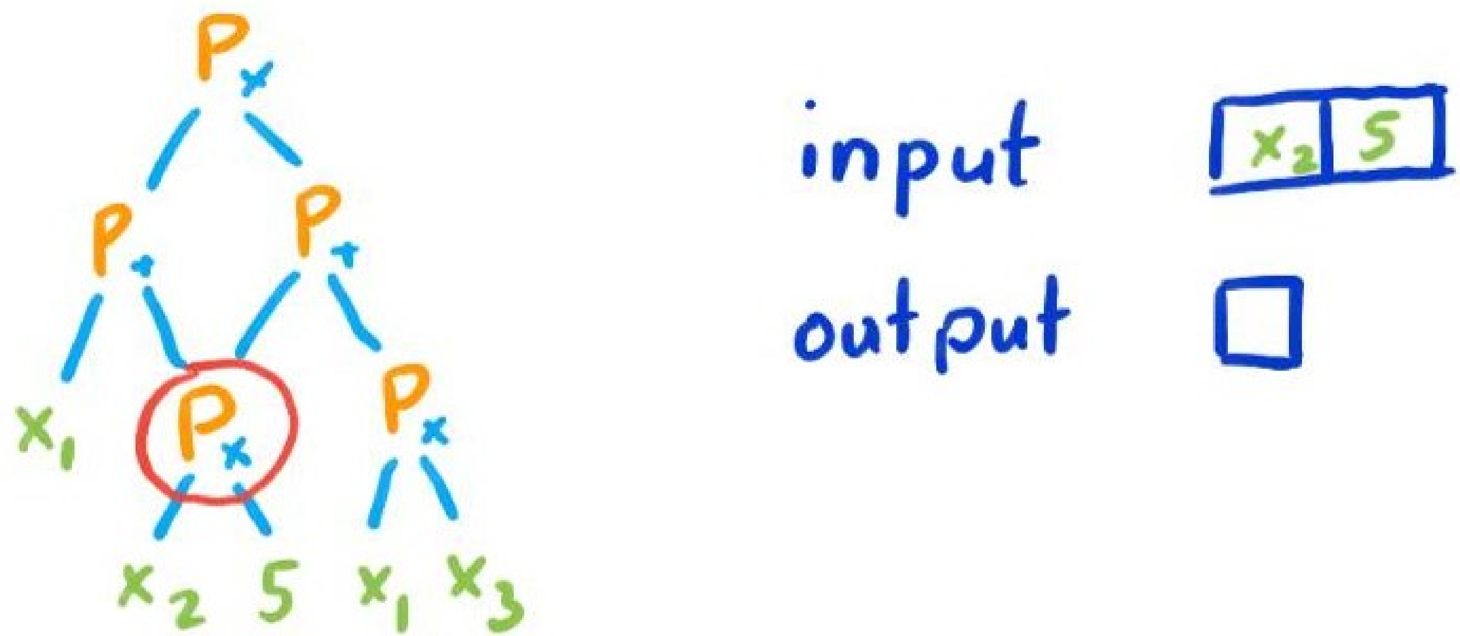
Theorem [BC'92]: $L \supseteq \#NC^1$

CLEAN COMPUTATION

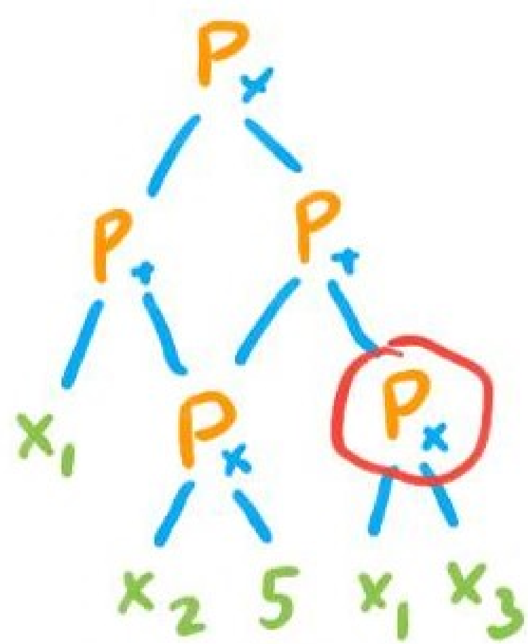


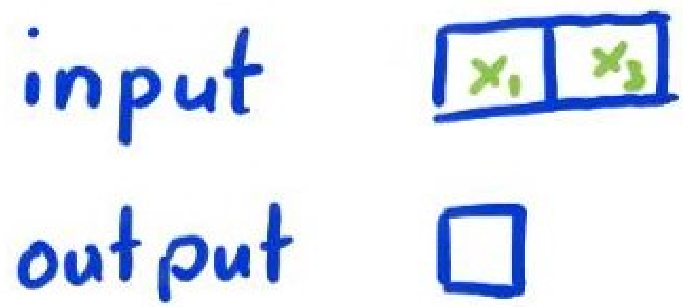


Clean Computation

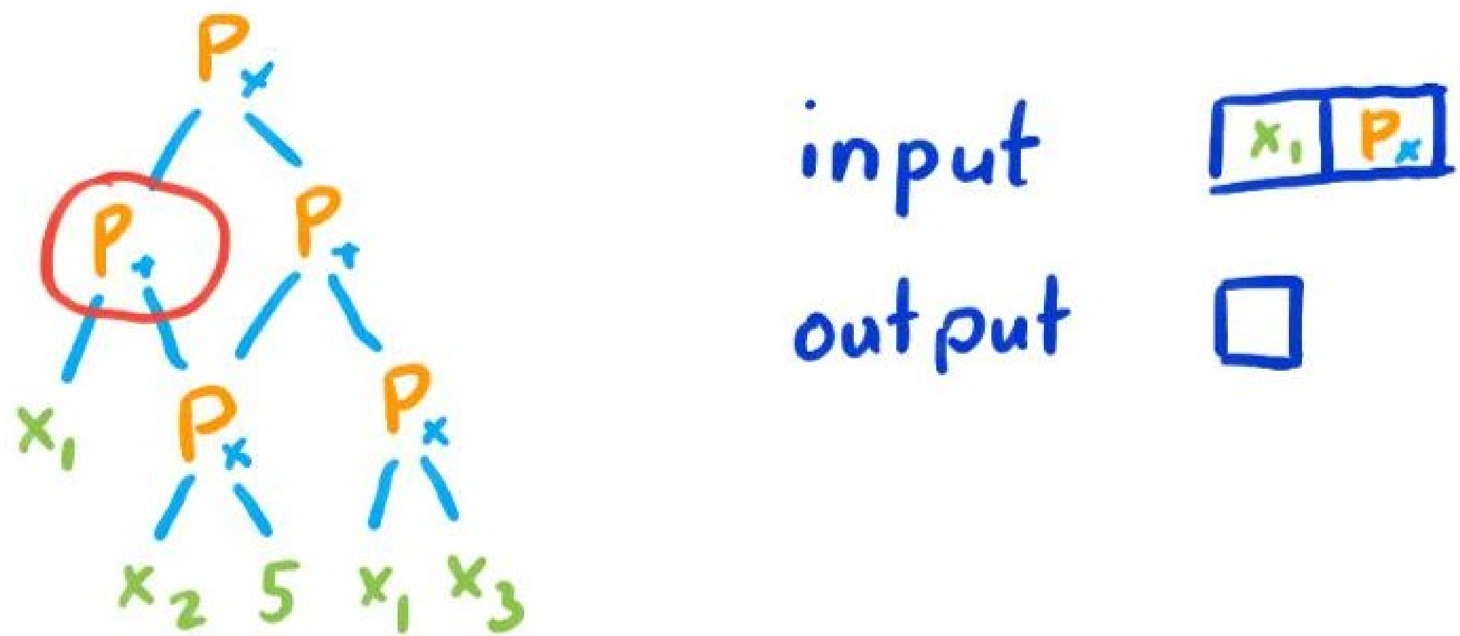


Clean Computation

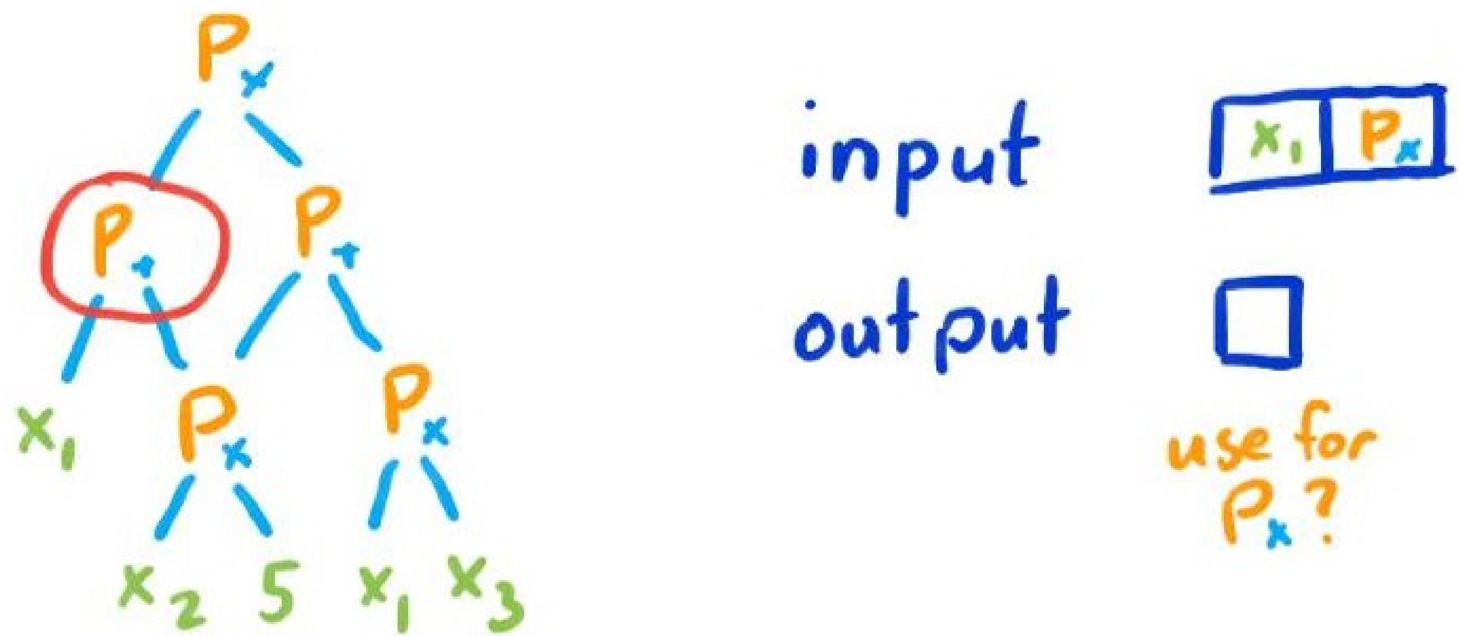




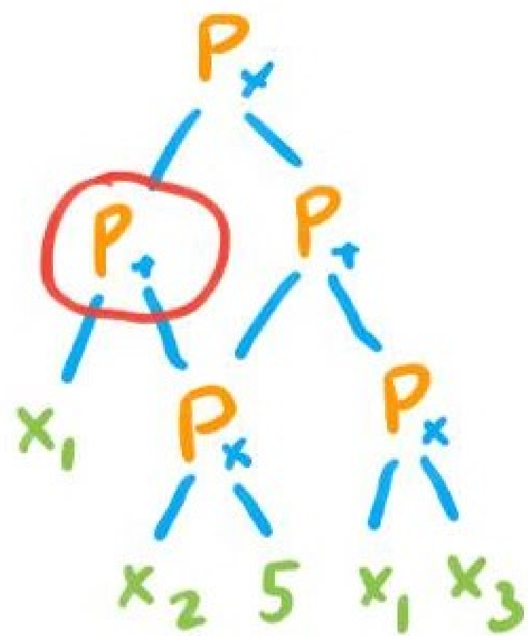
Clean Computation



CLEAN COMPUTATION



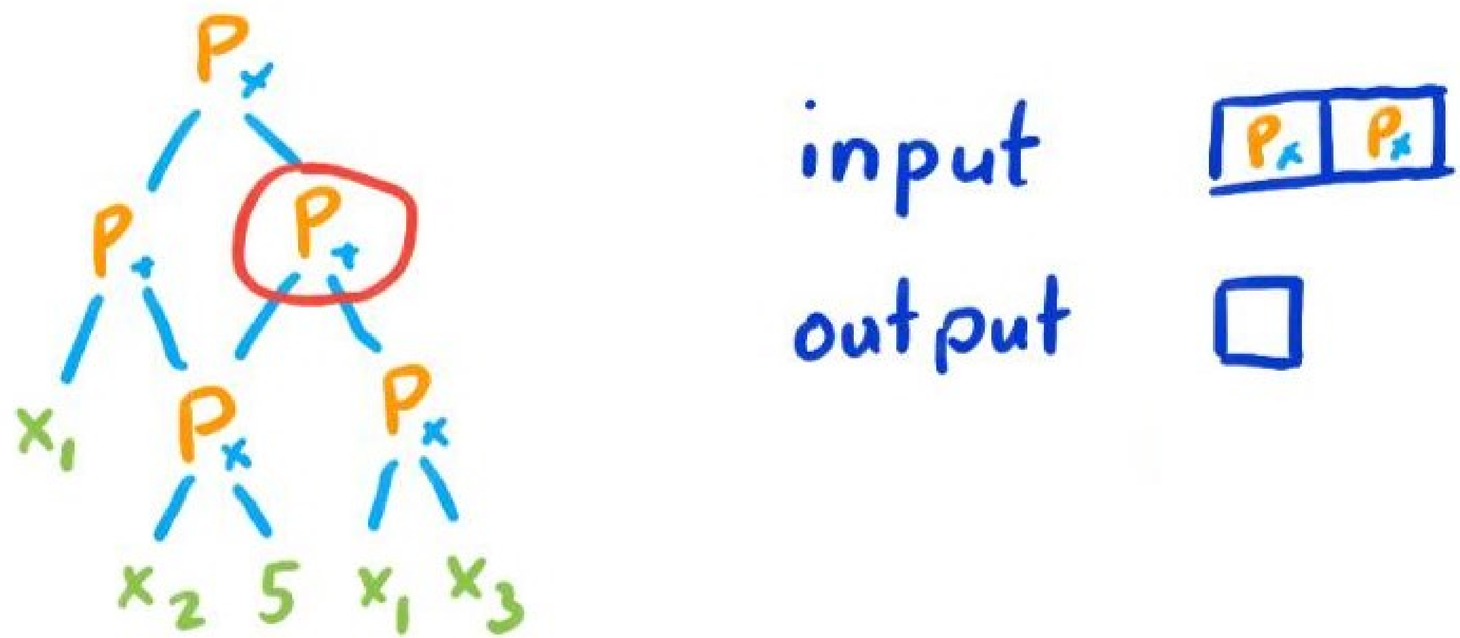
Clean Computation



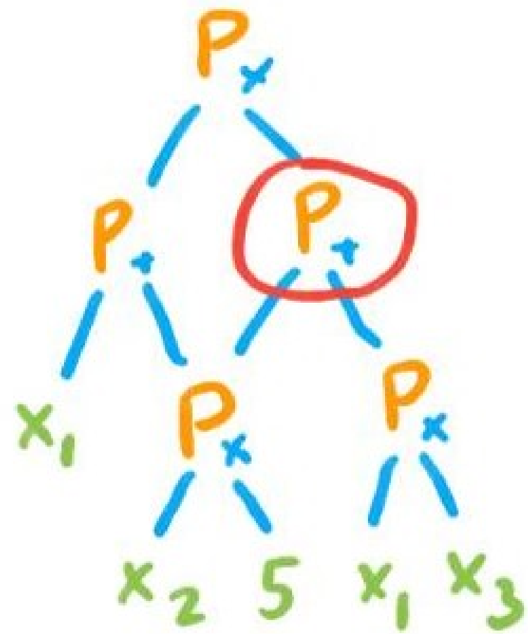
input $\boxed{x_2}\boxed{P_\times}$

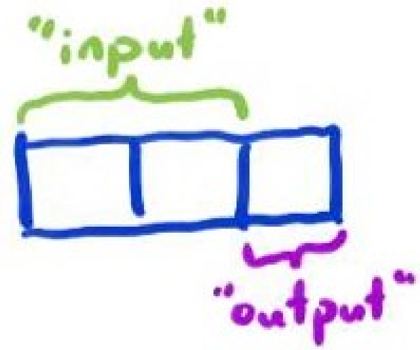
output $\boxed{5}$

Clean Computation

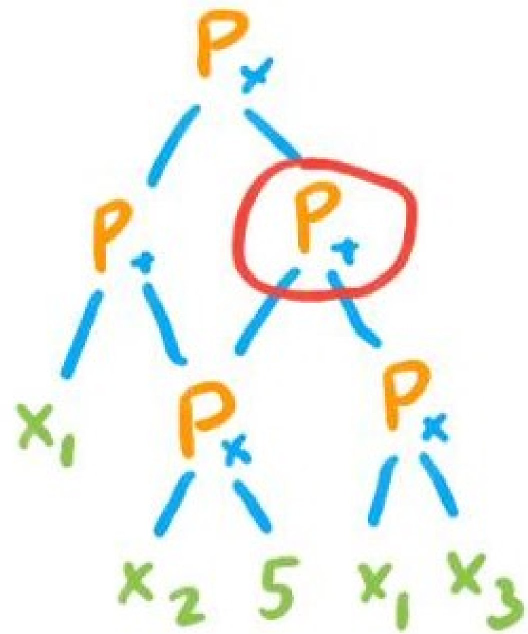


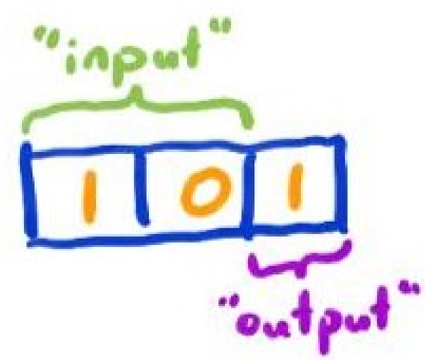
Clean Computation



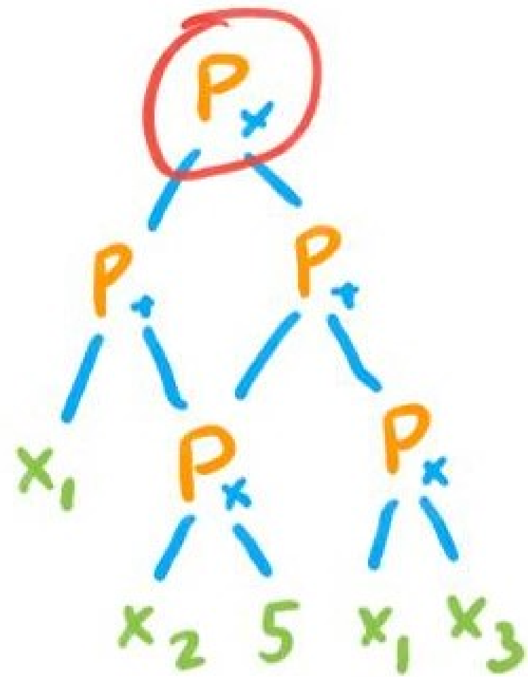


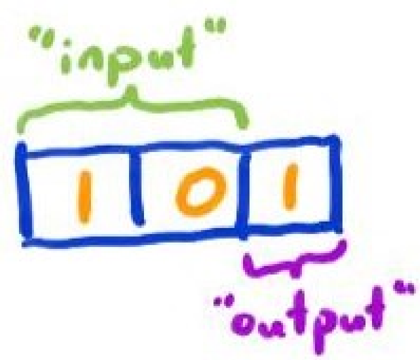
CLEAN COMPUTATION



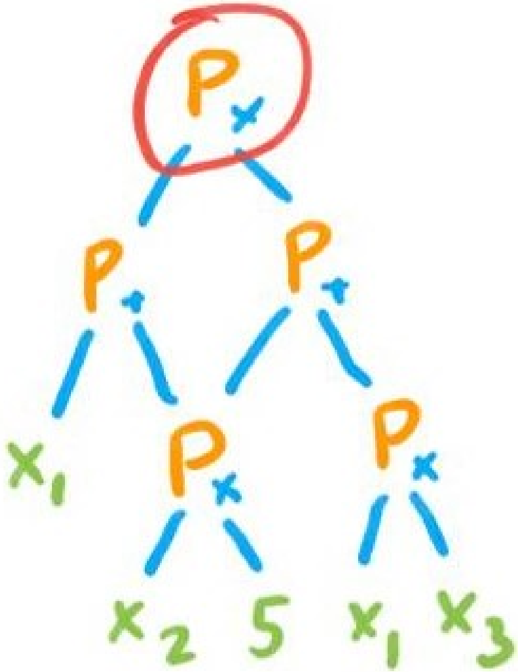


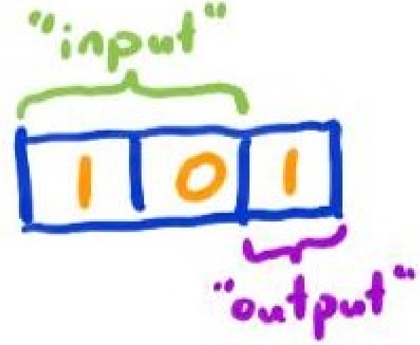
CLEAN COMPUTATION





Clean Computation





$$2^h = \text{poly}(n) \checkmark$$

CLEAN COMPUTATION
INPUT-MASKED

Clean Input-Masked* Computation

*patent pending

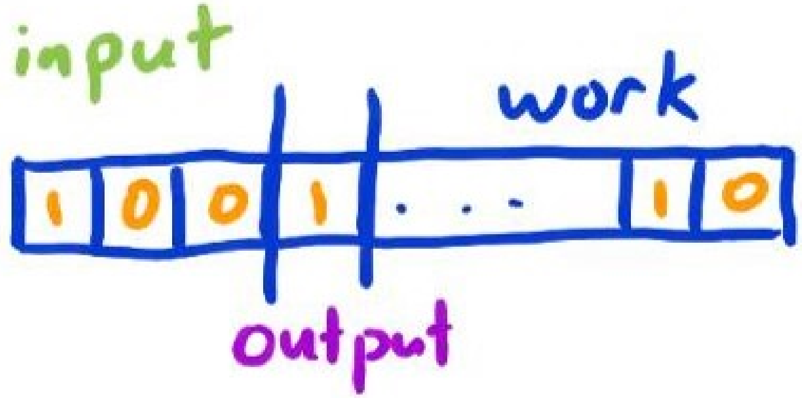
CLEAN COMPUTATION

INPUT-MASKED

~~input~~

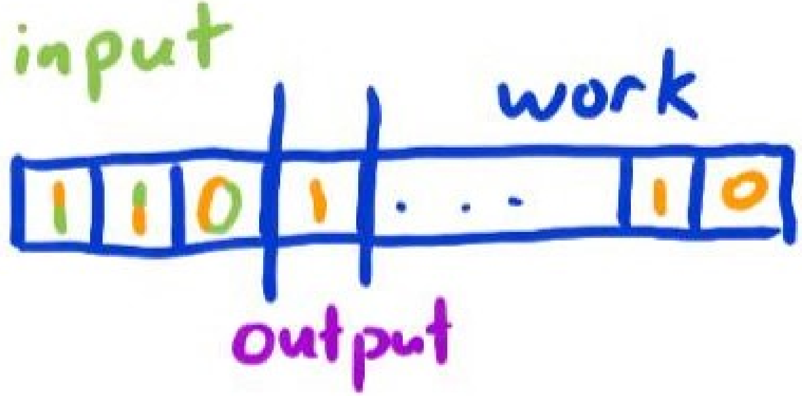
~~output~~

~~work~~



Clean Input-Masked Computation

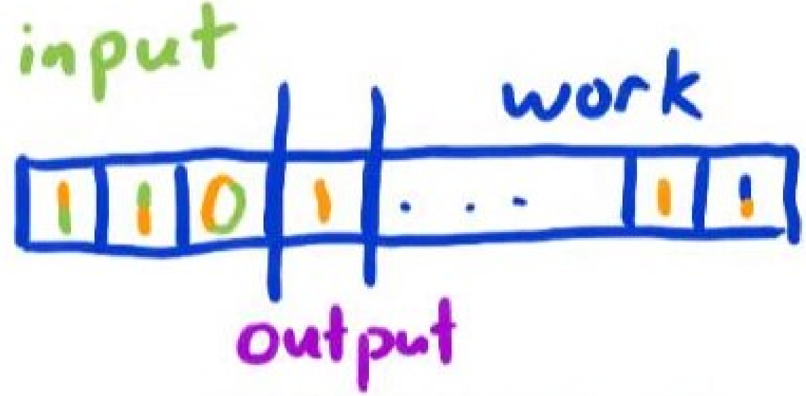
P_{in}



CLEAN COMPUTATION

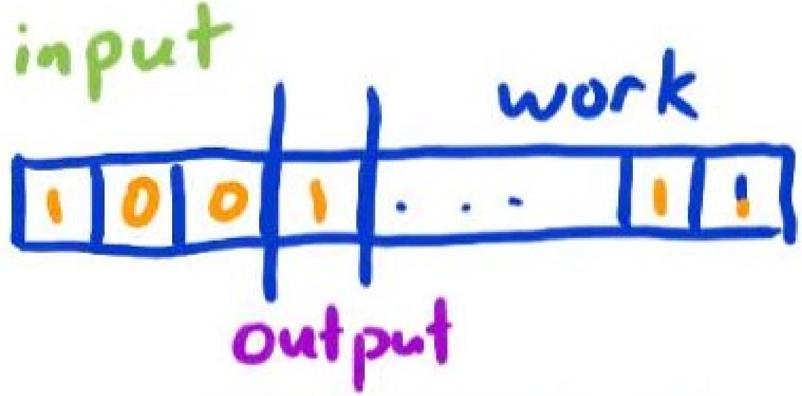
INPUT-MASKED

P_{in}



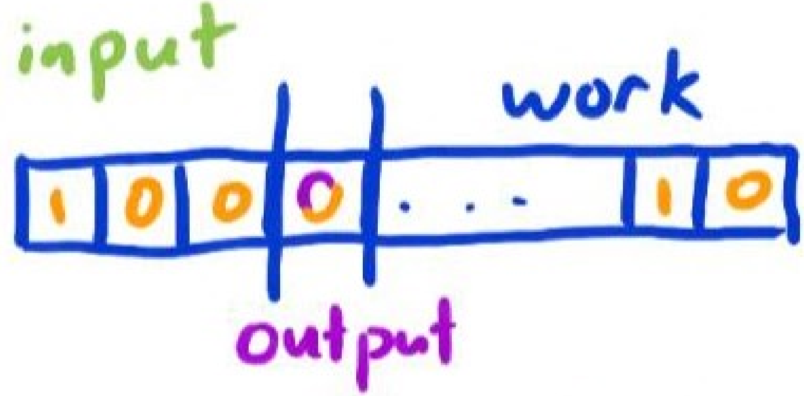
CLEAN INPUT-MASKED COMPUTATION

P_{in}



Clean Input-Masked Computation

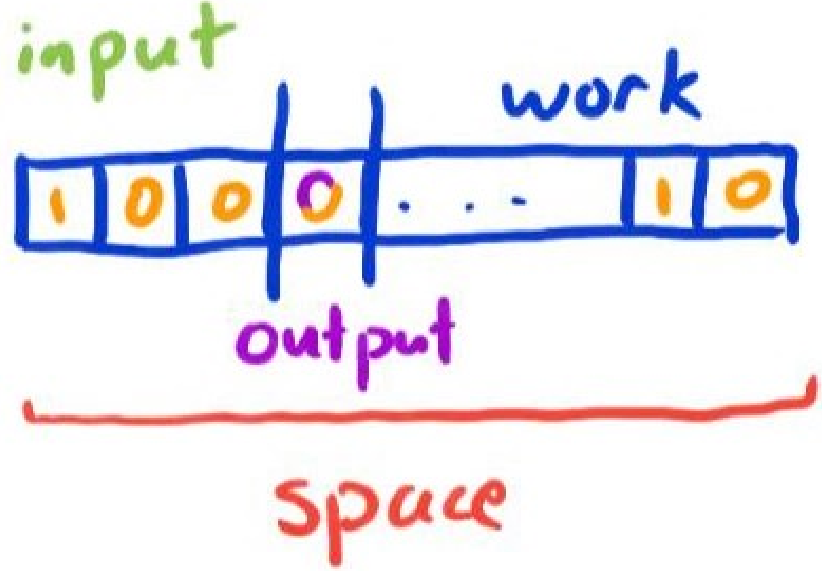
P_{in}



Clean Computation

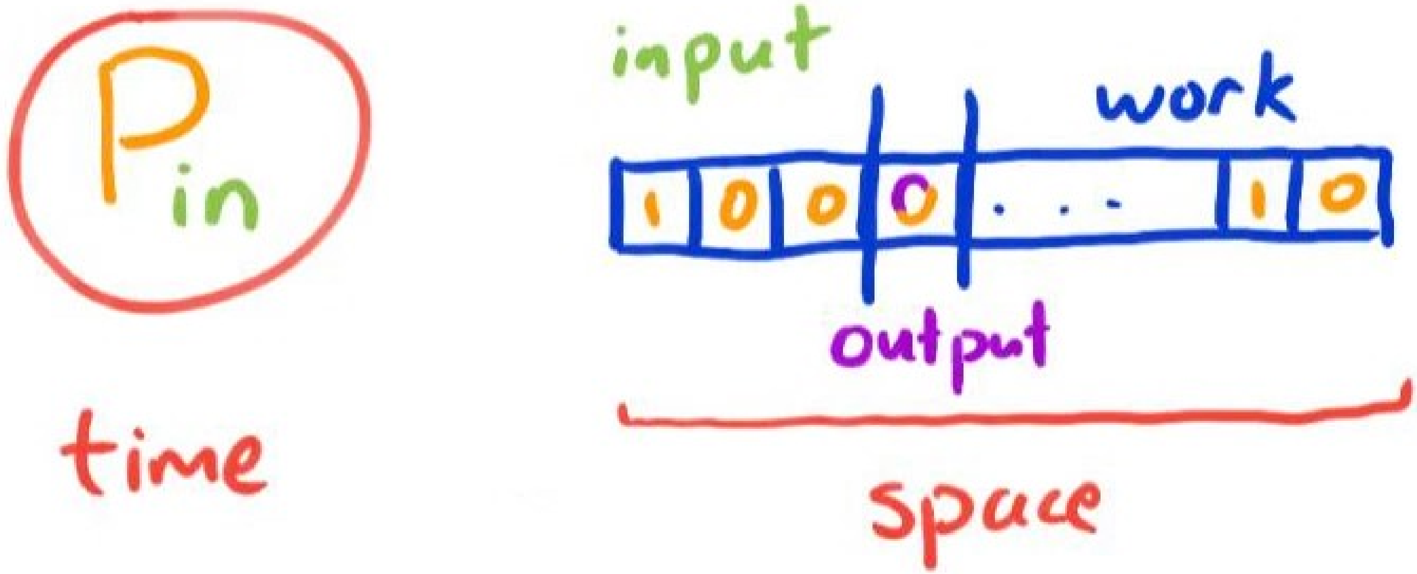
Input-Masked

P_{in}



CLEAN COMPUTATION

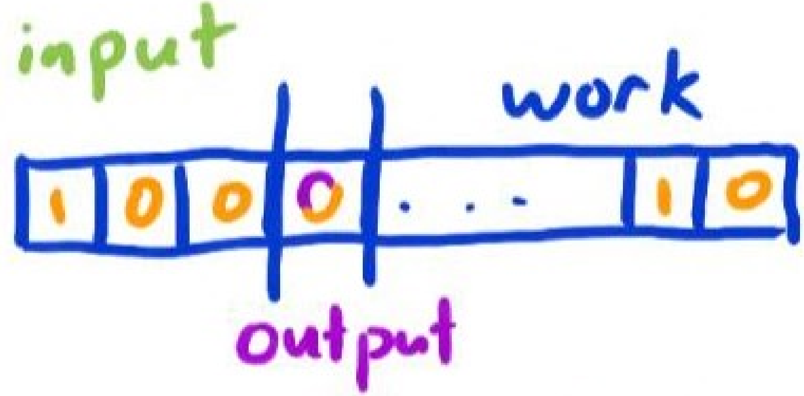
INPUT-MASKED



Clean Computation

Input-Masked

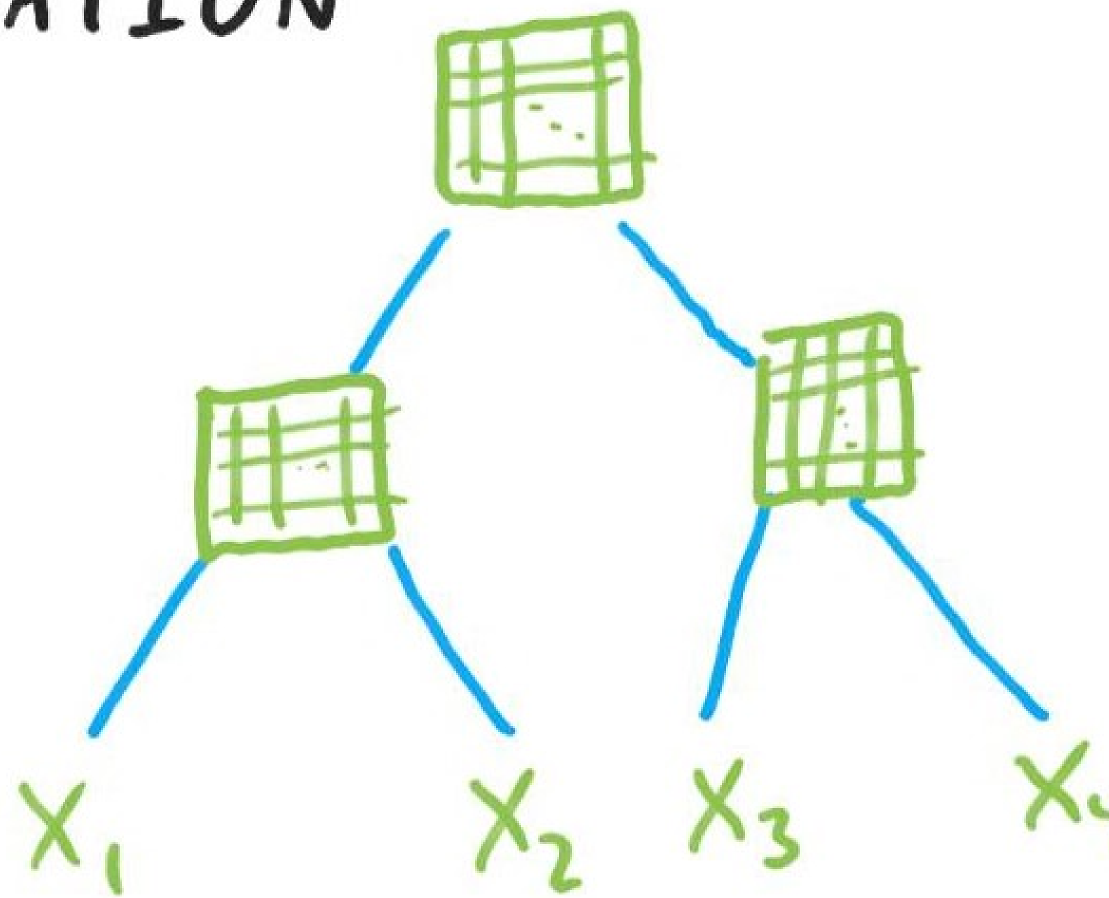
P_{in}



Log-Depth + Clean IM

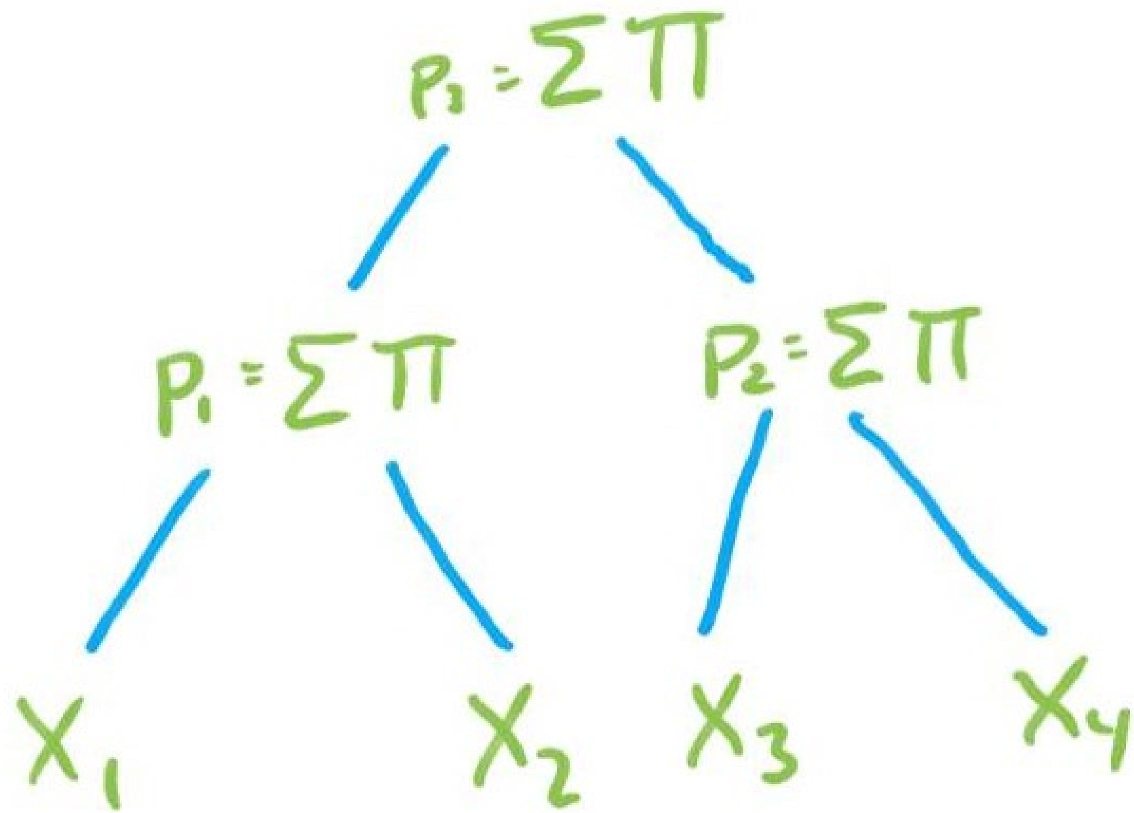
Tree Evaluation

height h
domain k



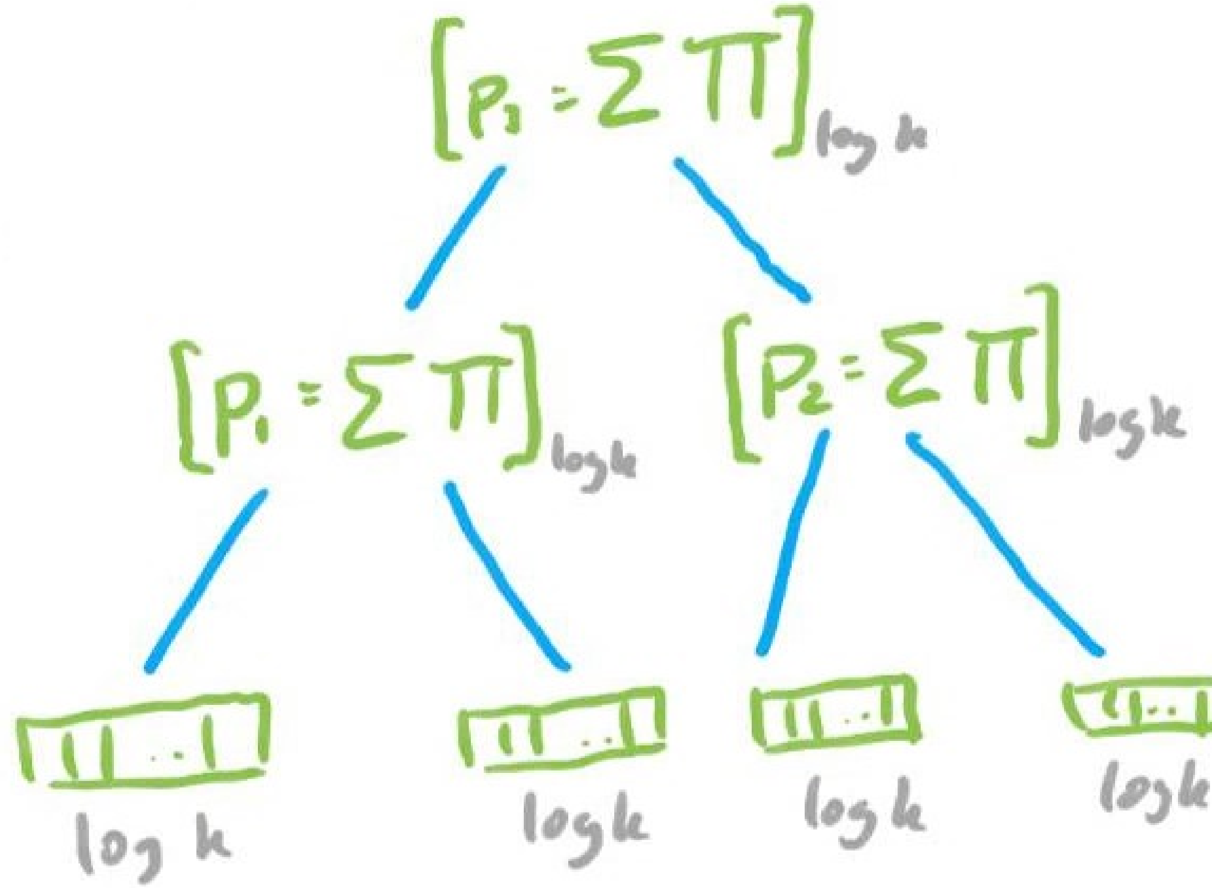
Tree Evaluation

height h
domain k



TREE EVALUATION

height h
domain k



Tree Evaluation

Lemma [CM'20]: d-ary product can be cleanly IM computed in space $d+1$ and time 2^d.

TREE EVALUATION

LEMMA [CM'20]: m degree d polynomials over n variables can be cleanly IM computed in space $n+m$ and time 2^d.

Tree Evaluation

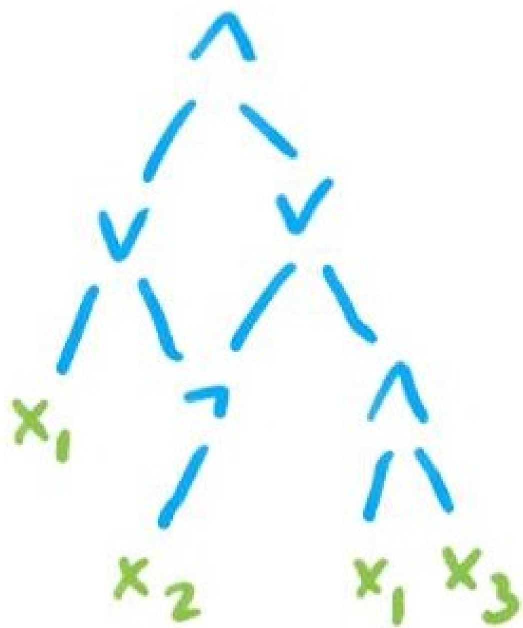
Proposal 1: $[\Sigma \Pi^d]_d$ is $(O(d), O(1))$-clean IM

TREE EVALUATION

LEMMA [CM'??]: d-ary product can be cleanly IM computed in space $O(d)$ and time $O(1)$.

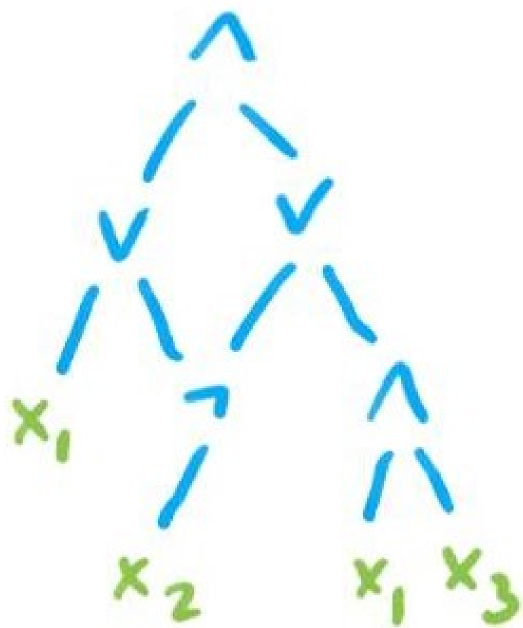
([CM'20]: so can all the sums)

CL vs NC^2



depth $\log^2$!

CL vs NC^2

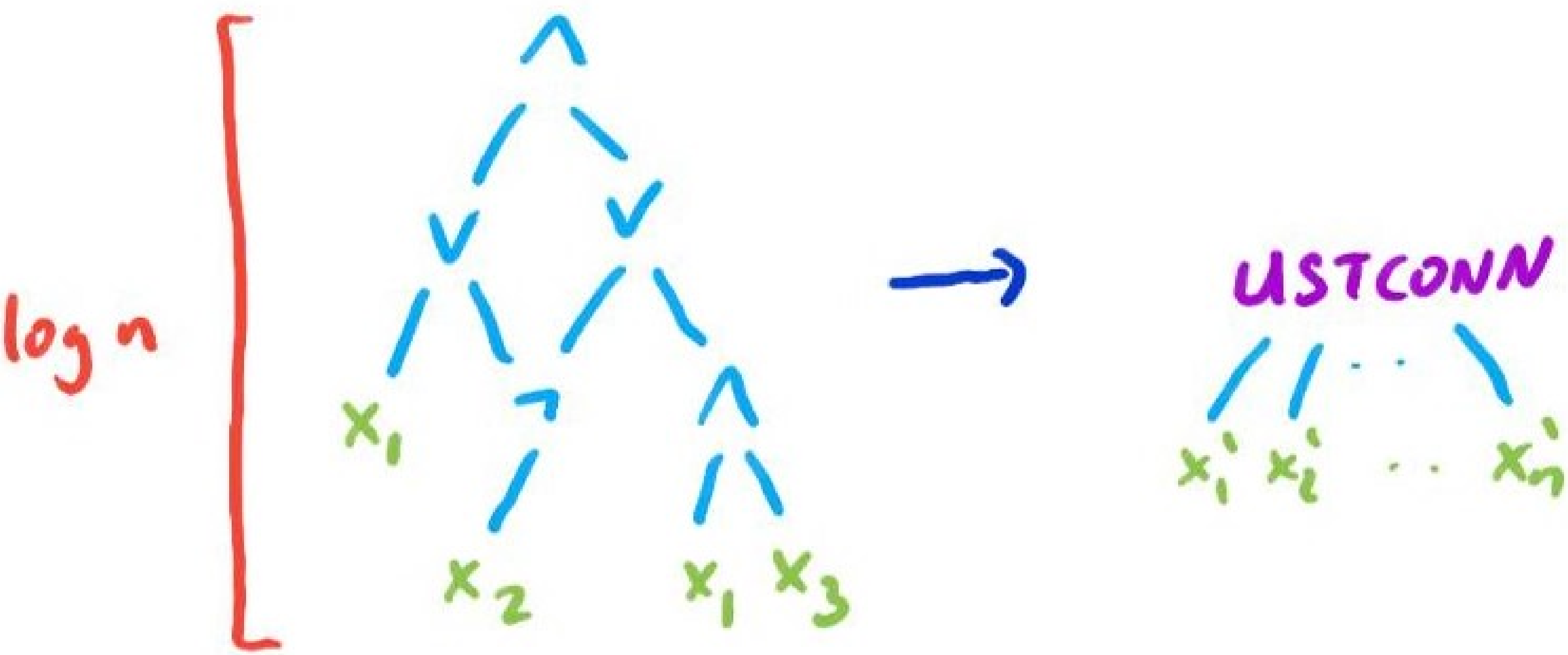


depth $\log^2$!

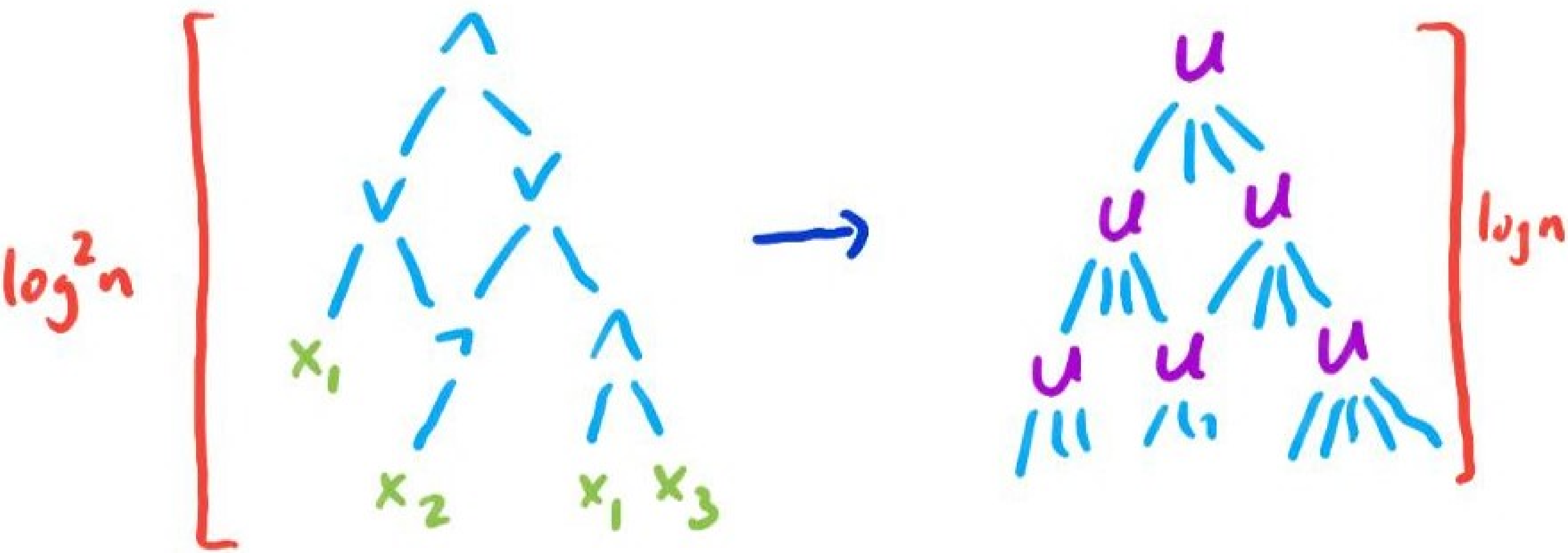
DEPTH IS HARD.

FUNCTIONS ARE EASY

CL vs NC²



CL vs NC^2



CL vs NC^2

PROPOSAL 2: $[\mathrm{USTCONN}]^{\mathrm{poly}(n)}$ is $(\mathrm{poly}(n), O(1))$-clean IM

CL vs NC^2

LEMMA [BCKLS'14]: powering to the d can be cleanly IM computed in space $O(d)$ and time $O(1)$ (over commutative rings).

OTHER OPTIONS

1) CL vs P
2) lifting with sunflowers

THANKS!