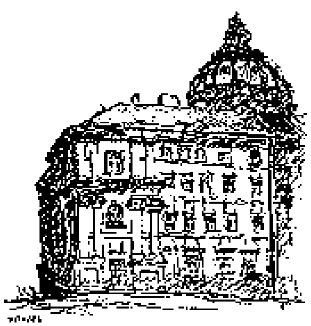
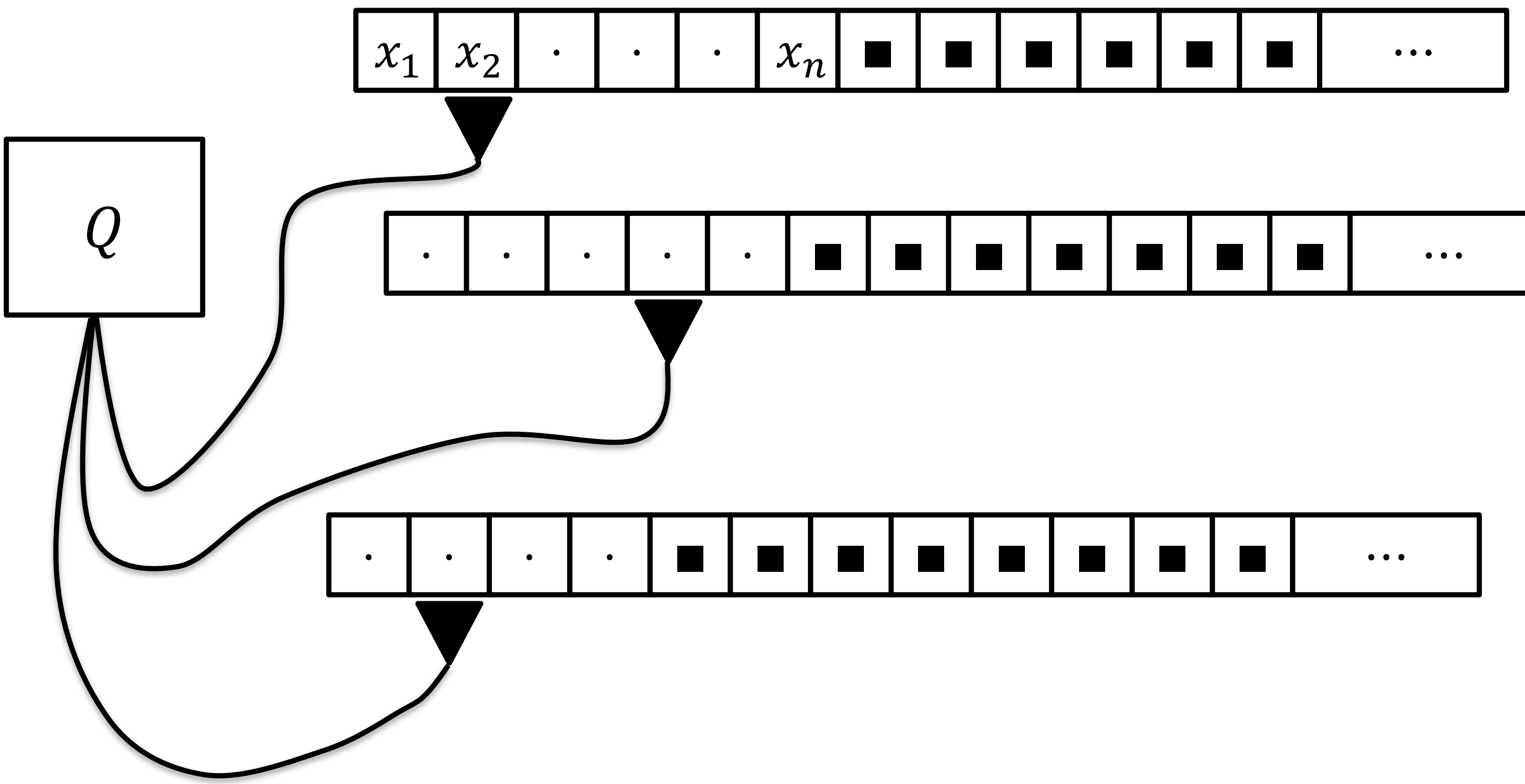
Journey with Eric Allender: From Turing machines to circuits, and back

Michal Koucký

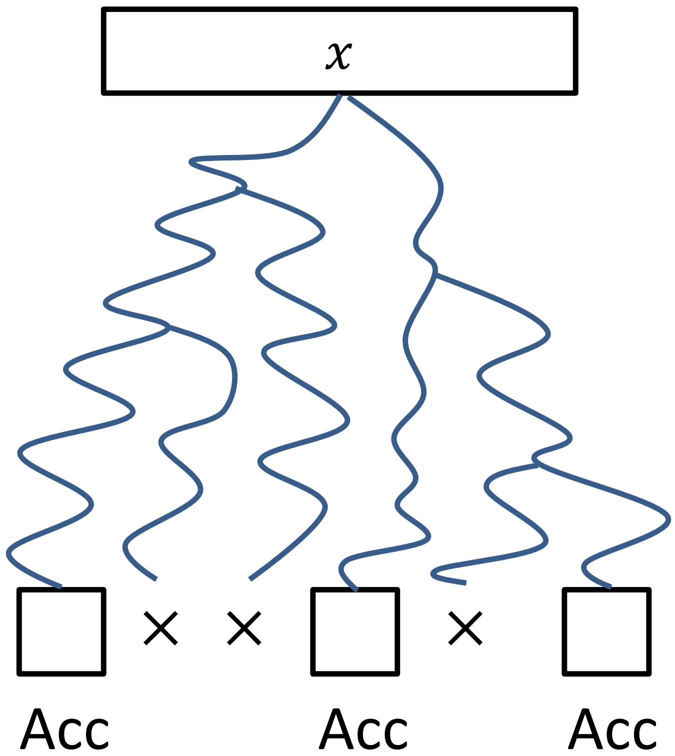
Charles University



Turing machine



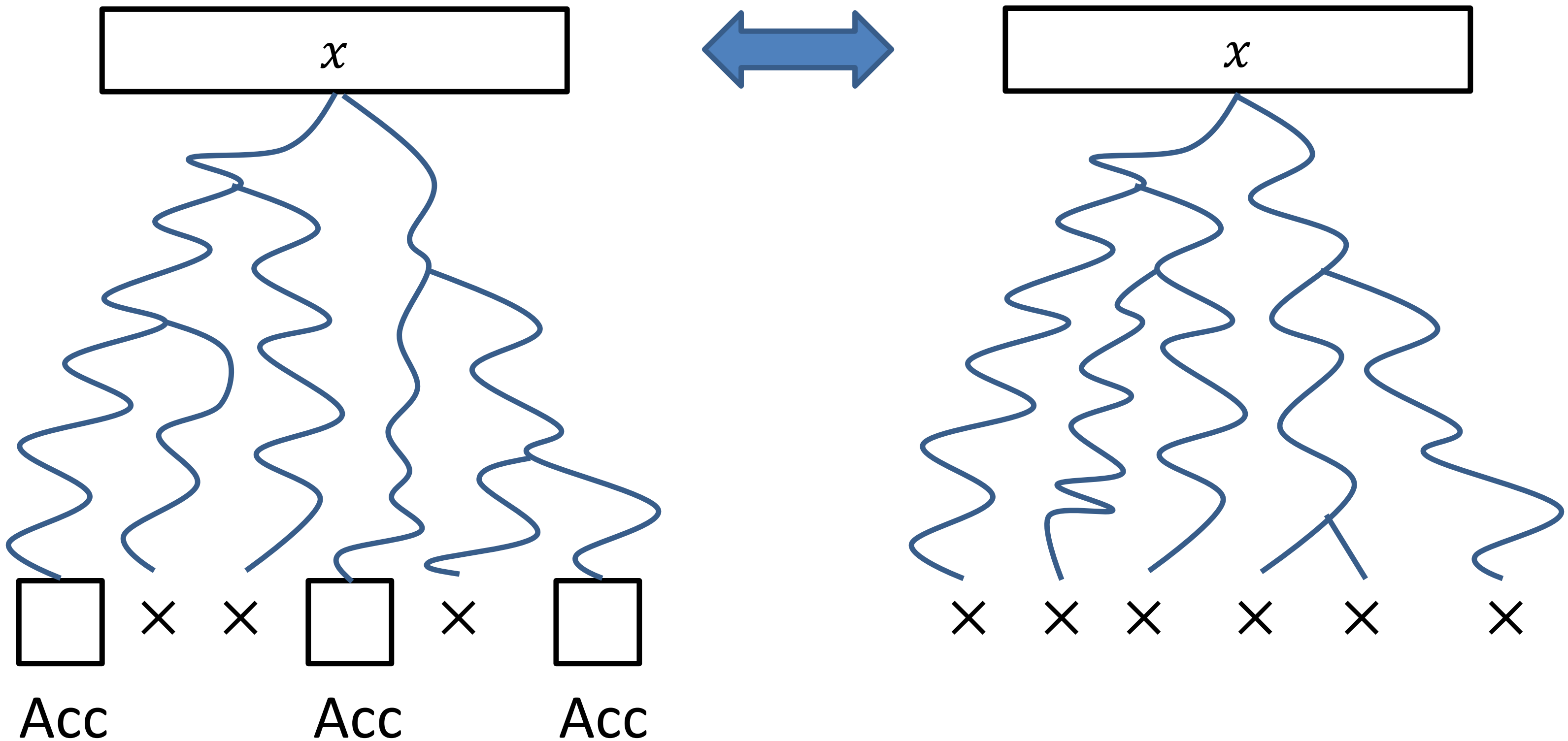
Nondeterministic computation



Computation tree

Co-nondeteministic computation

NP = coNP?



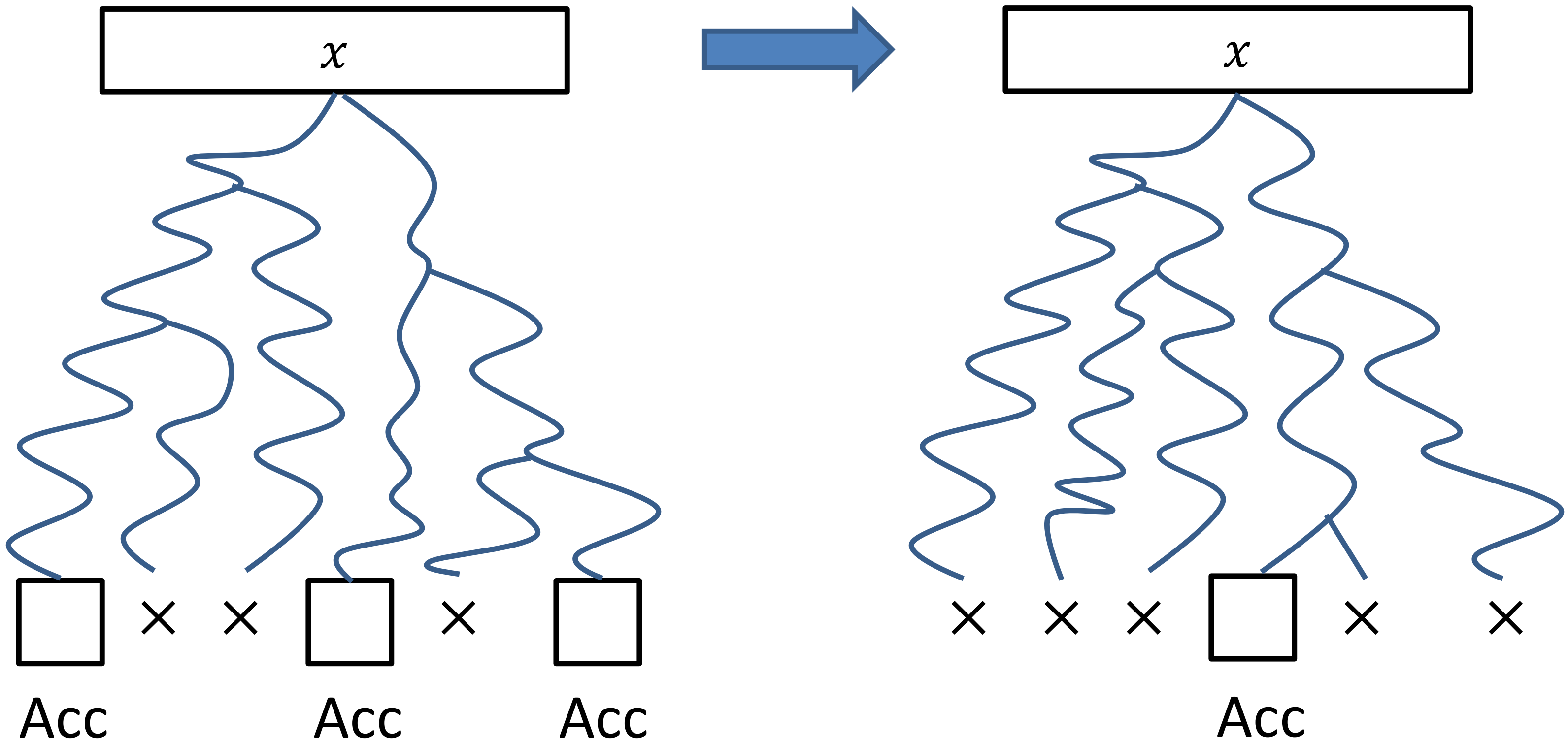
$x \in L$ iff $\exists$ accepting computation

$x \in L$ iff $\nexists$ accepting computation

Unambiguous computation

[Valiant-Vazirani'86]

"NP ⊆ UP/poly"



$x \in L$ iff $\exists$ accepting computation

$x \in L$ iff $\exists$ unique accepting computation

Unambiguous computation

[Valiant-Vazirani'86]

"NP $\subseteq$ UP/poly"

formula ϕ $\rightarrow$ formula ϕ'

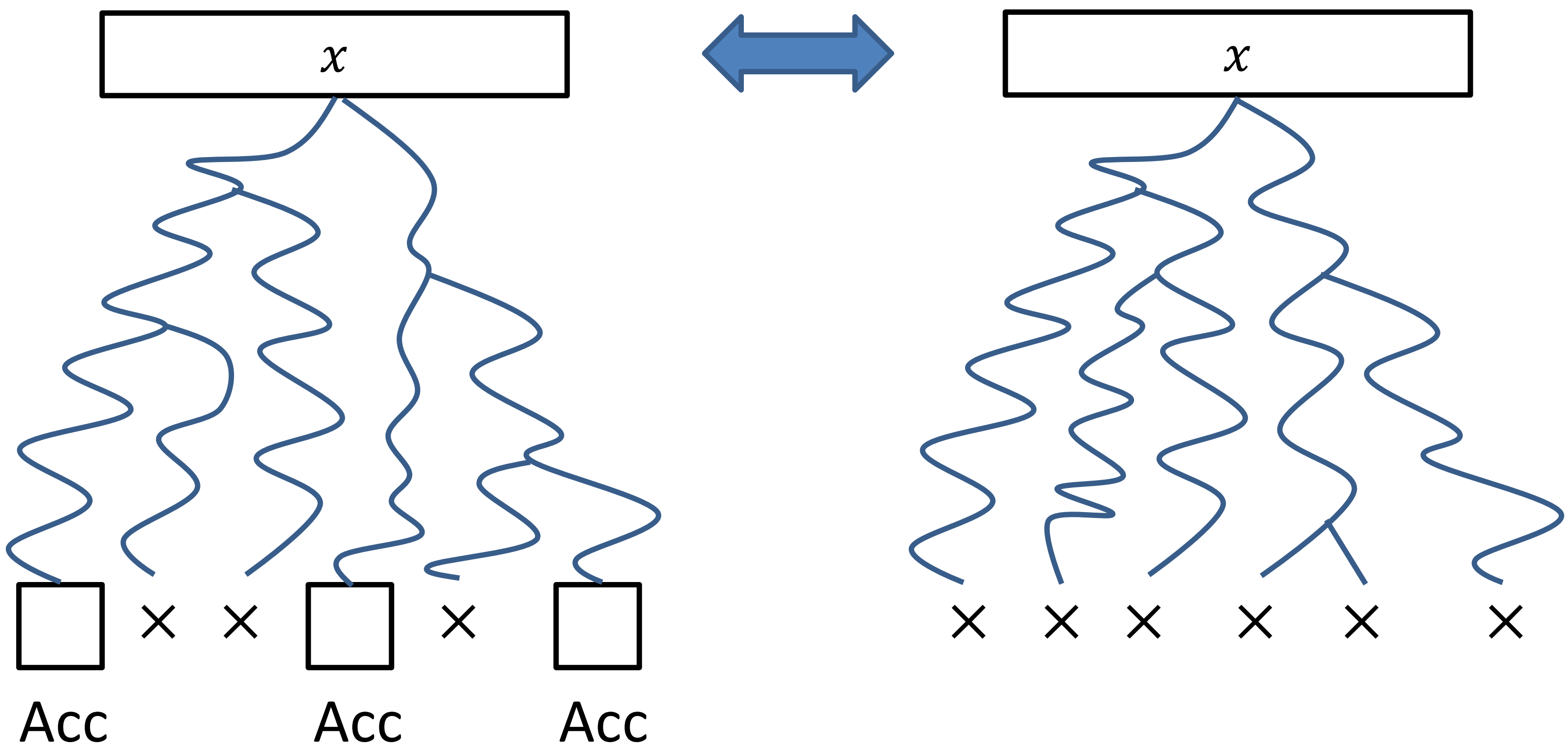
ϕ is satisfiable then ϕ' has unique satisfying assignment

ϕ is unsatisfiable then ϕ' is unsatisfiable

Co-nondeteministic computation

[Immerman-Szelepscényi'87]

NL = coNL



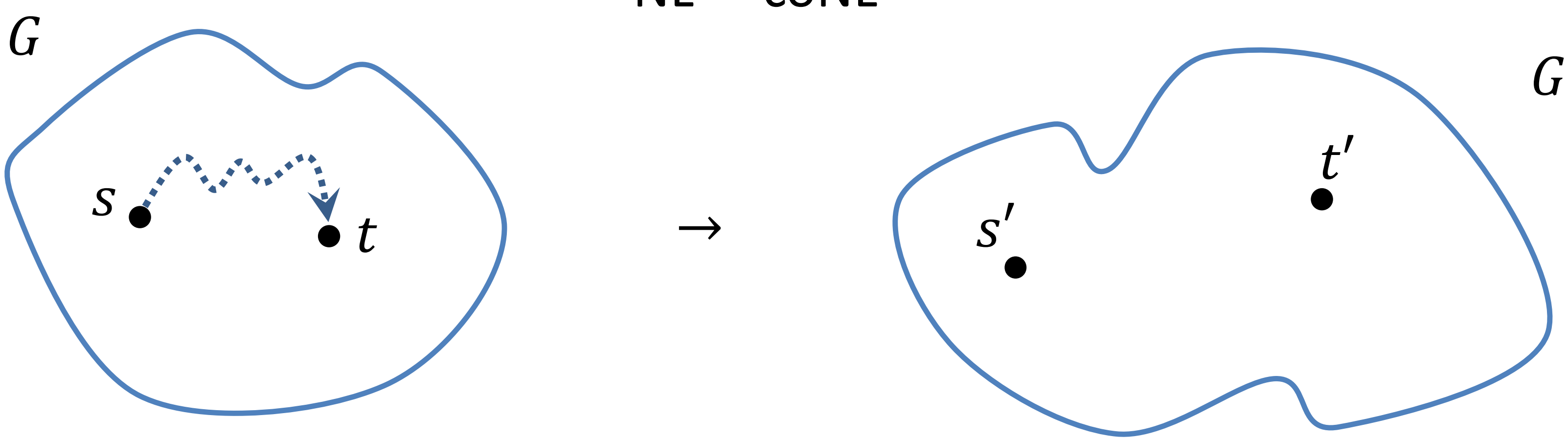
$x \in L$ iff $\exists$ accepting computation

$x \in L$ iff $\nexists$ accepting computation

Co-nondeteministic computation

[Immerman-Szelepscényi'87]

NL = coNL

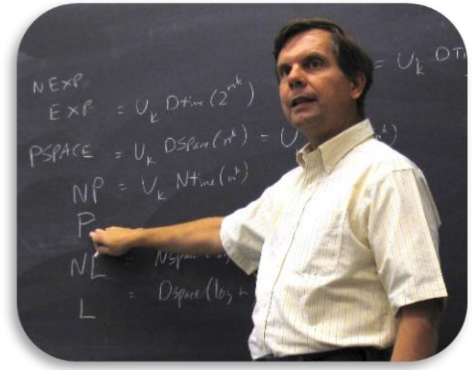


(G, s, t) $\rightarrow$ (G', s', t')

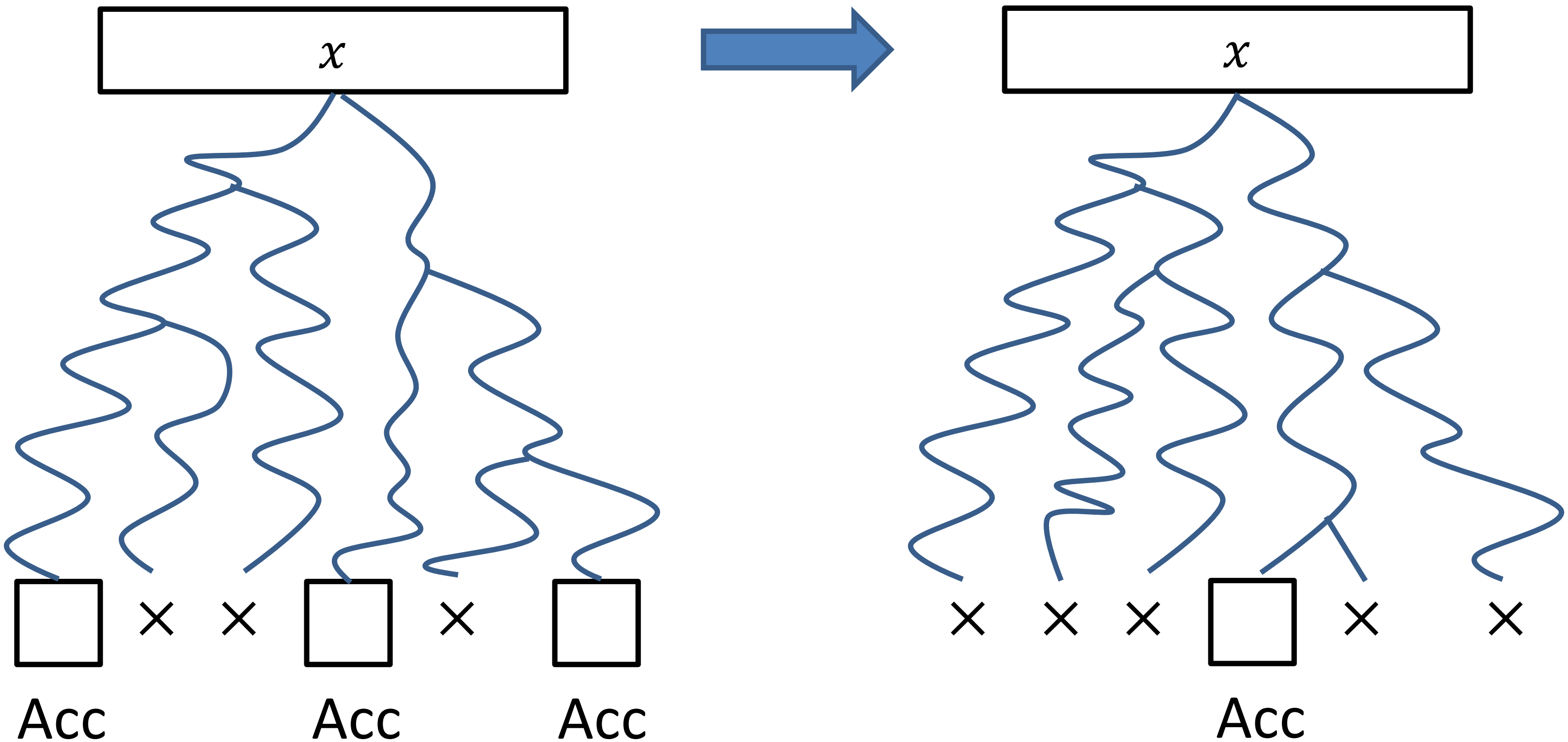
path from s to t in G iff no path from s' to t' in G'

Unambiguous computation

[Allender-Reinhardt'00]



NL ⊆ UL/poly

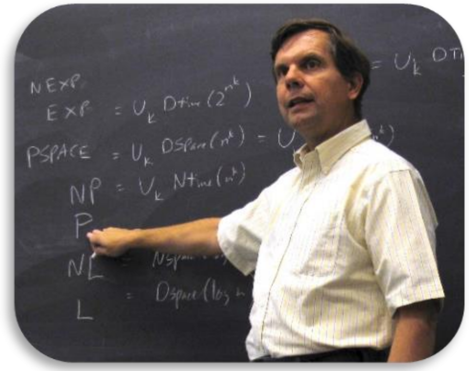


$x \in L$ iff ∃ accepting computation

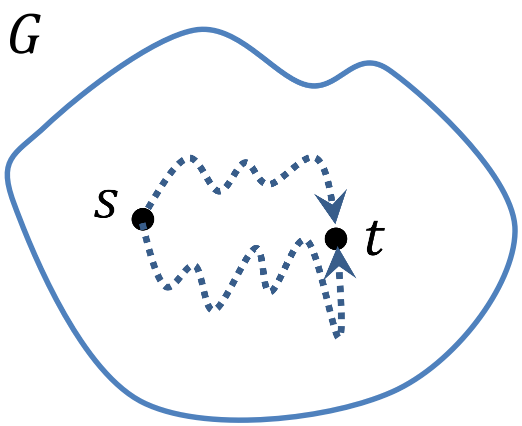
$x \in L$ iff ∃ unique accepting computation

Unambiguous computation

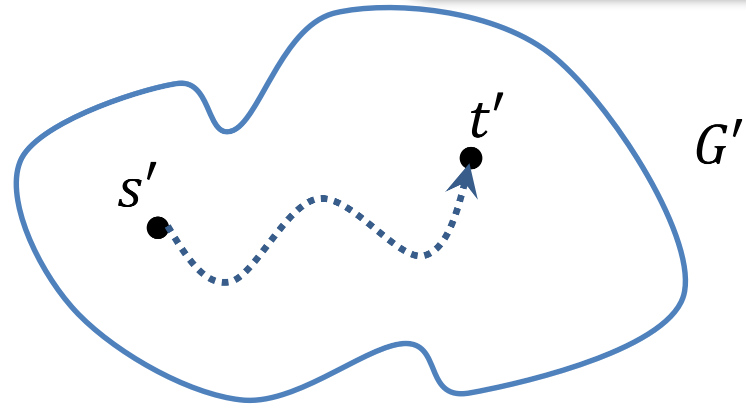
[Allender-Reinhardt'00]



NL ⊆ UL/poly



$\to$

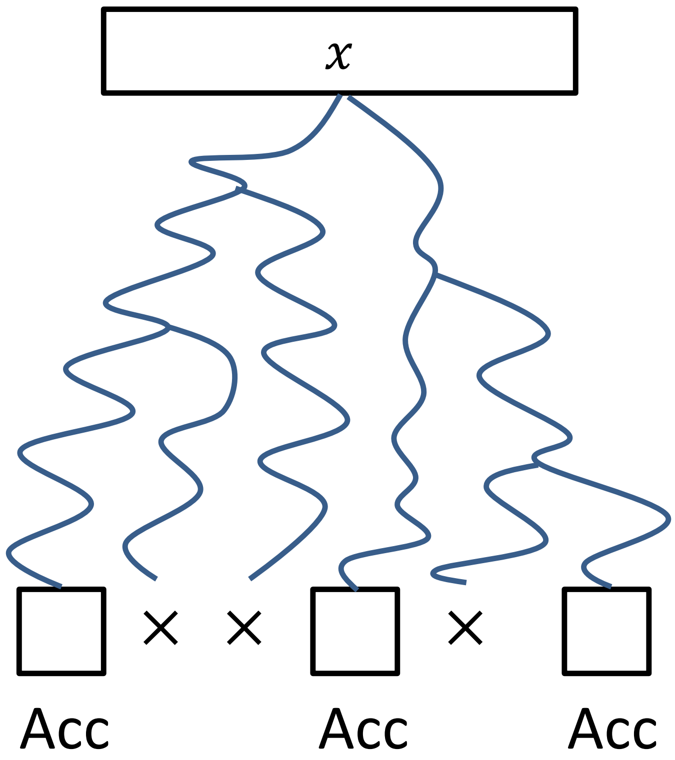


(G, s, t) $\to$ (G', s', t')

path from s to t in G then unique path from s' to t' in G'

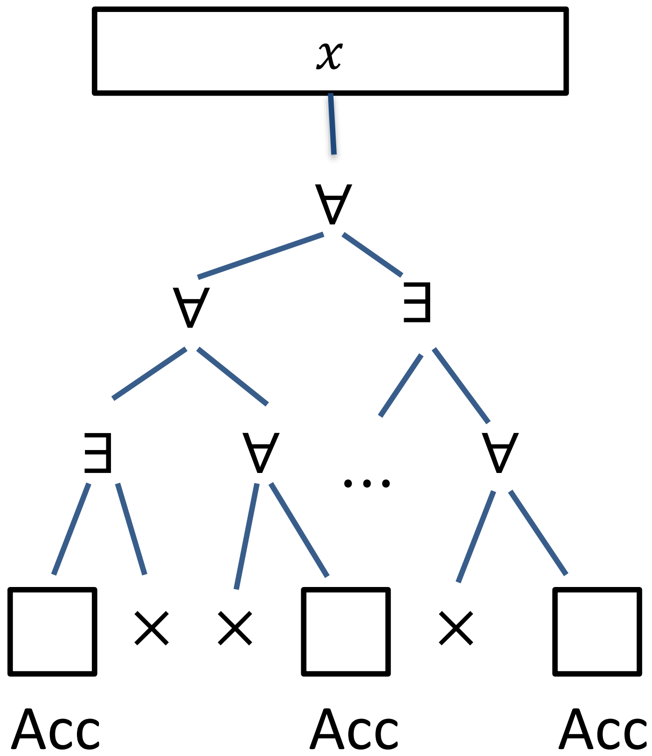
no path from s to t in G then no path from s' to t' in G'

Nondeterministic computation



Computation tree

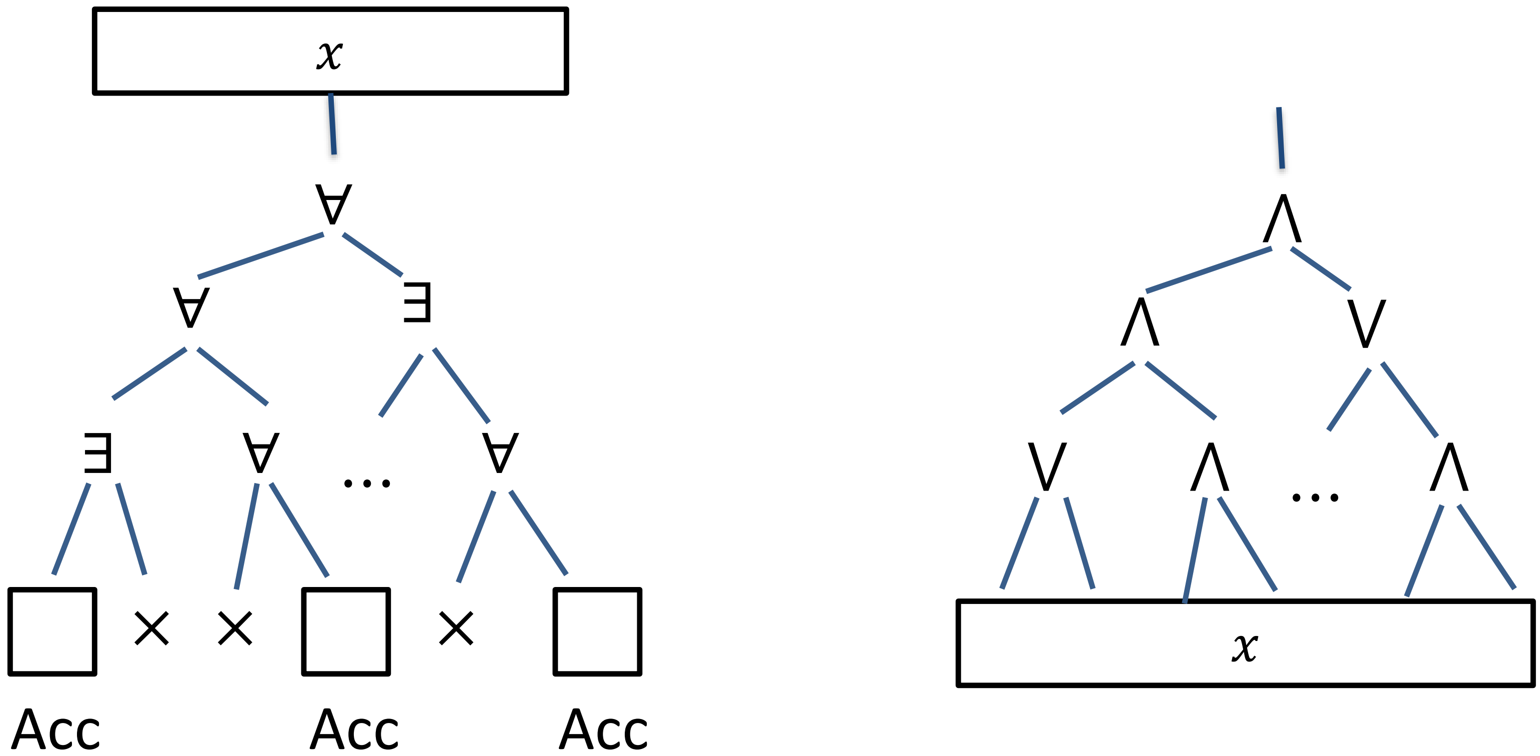
Alternating computation



Computation tree

Alternating computation → circuits

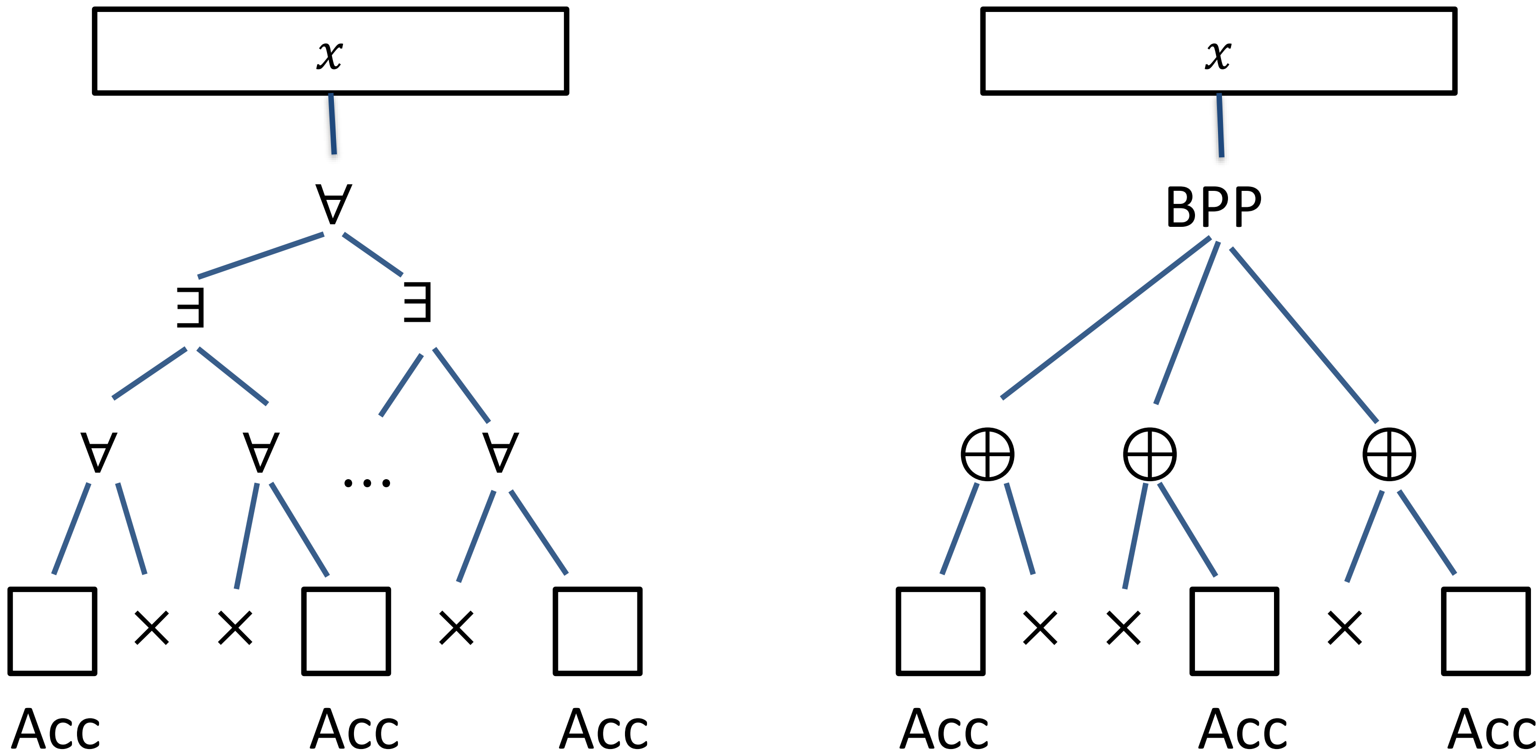
[Ruzzo'80]



Polynomial Hierarchy → exp. size AC^0 circuits

Polynomial Hierarchy in $\mathrm{BPP}^{\oplus\mathrm{P}}$

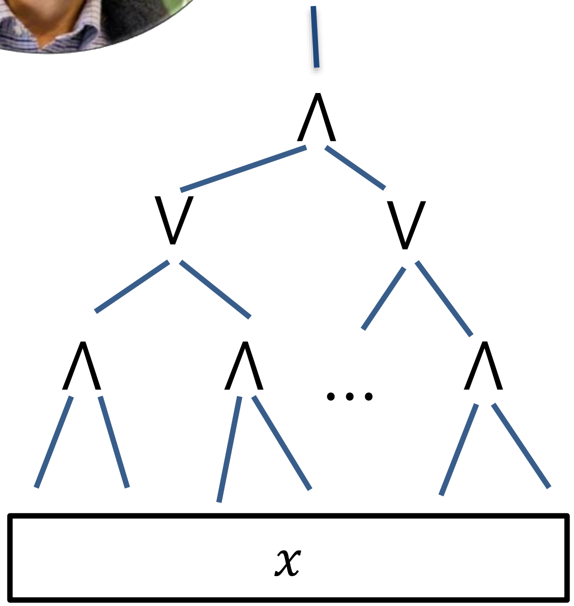
[Toda'89]

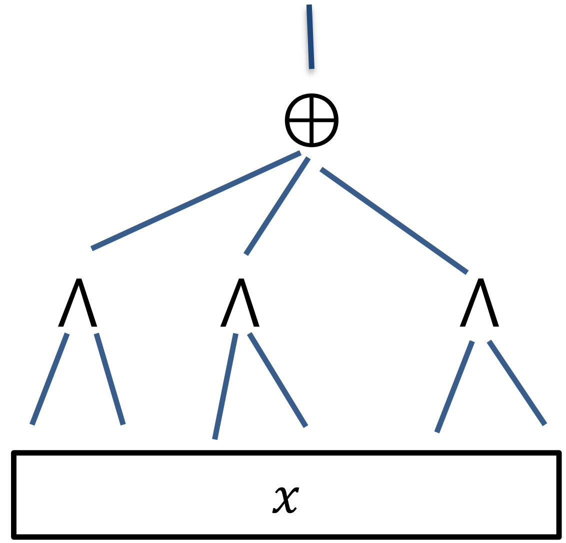


Polynomial Hierarchy → $\mathrm{BPP} \cdot \oplus\mathrm{P}$ computation

$AC^0[\oplus]$ in TC^0

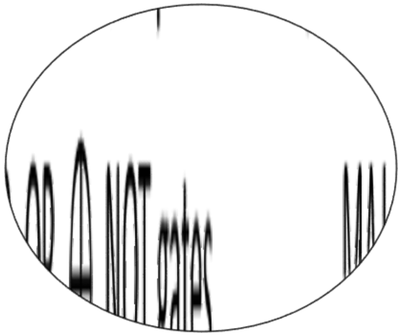
[Allender'89]





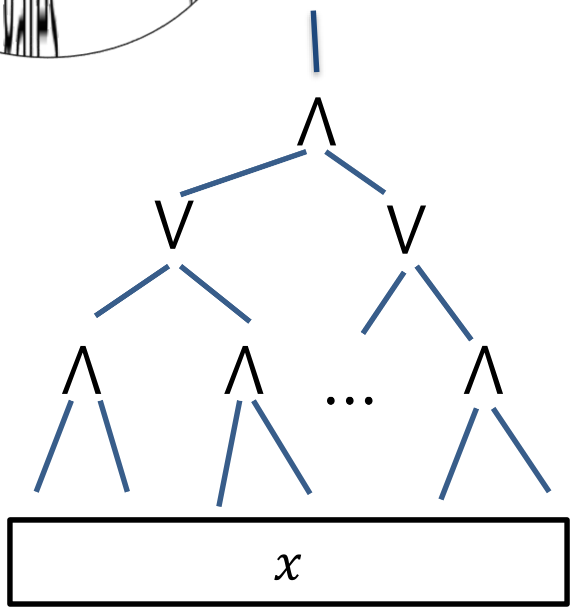
$AC^0[\oplus]$ = poly-size, $O(1)$-depth AND, OR, $\oplus$, NOT gates

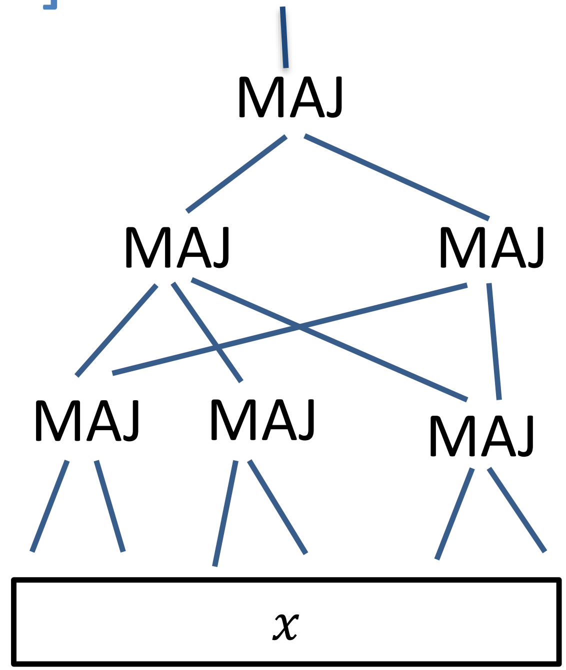
quasi-poly-size, 2-depth $\oplus \circ$ AND (probabilistic)



$AC^0[\oplus]$ in TC^0

[Allender'89]



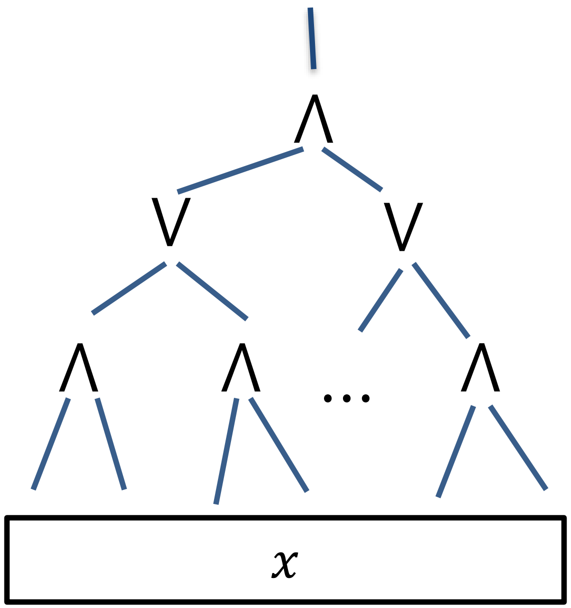


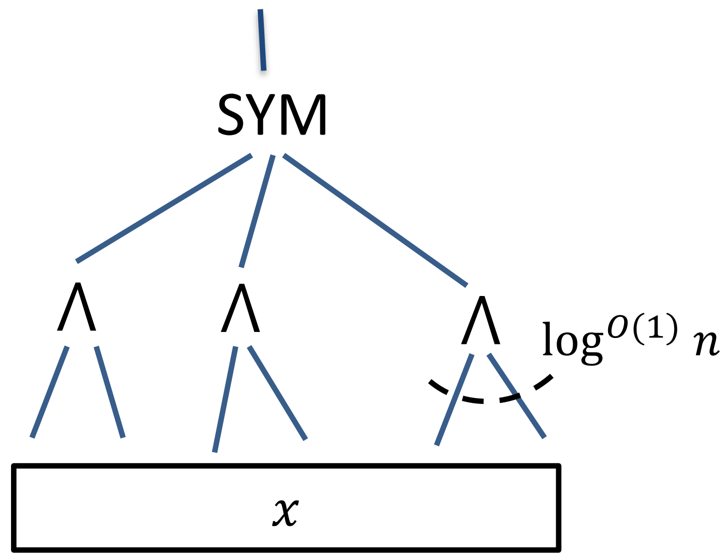
$AC^0[\oplus]$ = poly-size, $O(1)$-depth
AND, OR, $\oplus$, NOT gates

quasi-poly-size, 3-depth
MAJ, NOT gates

ACC^0 depth reduction

[Yao'90, Beigel-Tarui'94]





ACC^0 = poly-size, $O(1)$-depth AND, OR, MOD-m gates

quasi-poly-size, 2-depth AND gates and a SYM gate

Lower bounds

Allender-Gore'94: PERM $\notin$ uniform-ACC^0 of size $2^{n^{o(1)}}$

Allender'99: PERM $\notin$ uniform-TC^0 of size $2^{\log^{o(1)} n}$

Williams'14: NEXP $\notin$ ACC^0 of size $2^{n^{o(1)}}$

Permanent vs Determinant

PERM

#P-complete

DET

$\approx$ #L

#L $\subseteq$ NC^3

BMM $\in$ NC^1 — $A \times A$ over Boolean semiring

CONN $\in$ NC^2 — A^n over Boolean semiring

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

NC^i binary AND, OR, NOT

poly-size, $\log^i n$-depth

Small depth circuits

Question: $ACC^0 = TC^0$?

Question: $TC^0 = NC^1$?

Question: PERM in uniform-NC^3 ?

Known: $ACC^0 \subseteq TC^0$

$TC^0 \subseteq NC^1$

DET in uniform-NC^3

Allender-me'10: If MAJ is in ACC^0 then MAJ has ACC^0 ckt's of size $n^{1+\varepsilon}$.

If FLE is in TC^0 then FLE has TC^0 ckt's of size $n^{1+\varepsilon}$.

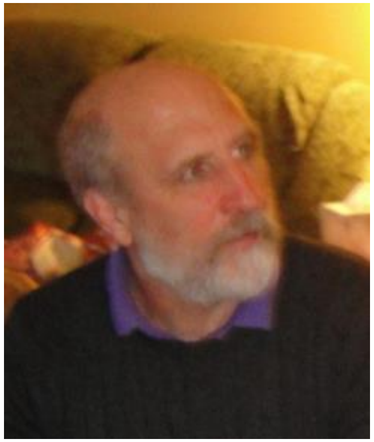
Chen-Tell'19: If MAJ is in ACC^0 then MAJ has ACC^0 ckt's of size n^{1+1/c^d}.

If FLE is in TC^0 then FLE has TC^0 ckt's of size n^{1+1/c^d}.

Arithmetic functions

- ADD, SUM $\in AC^0$ $x + y \to z$ $2n$ bits $\to n + 1$ bits
- MULT $\in TC^0$ $x * y \to z$ $2n$ bits $\to n^2$ bits
- DIV $\in TC^0$ Allender-Barrington-Hesse'02

Question: DET $\in TC^0$?

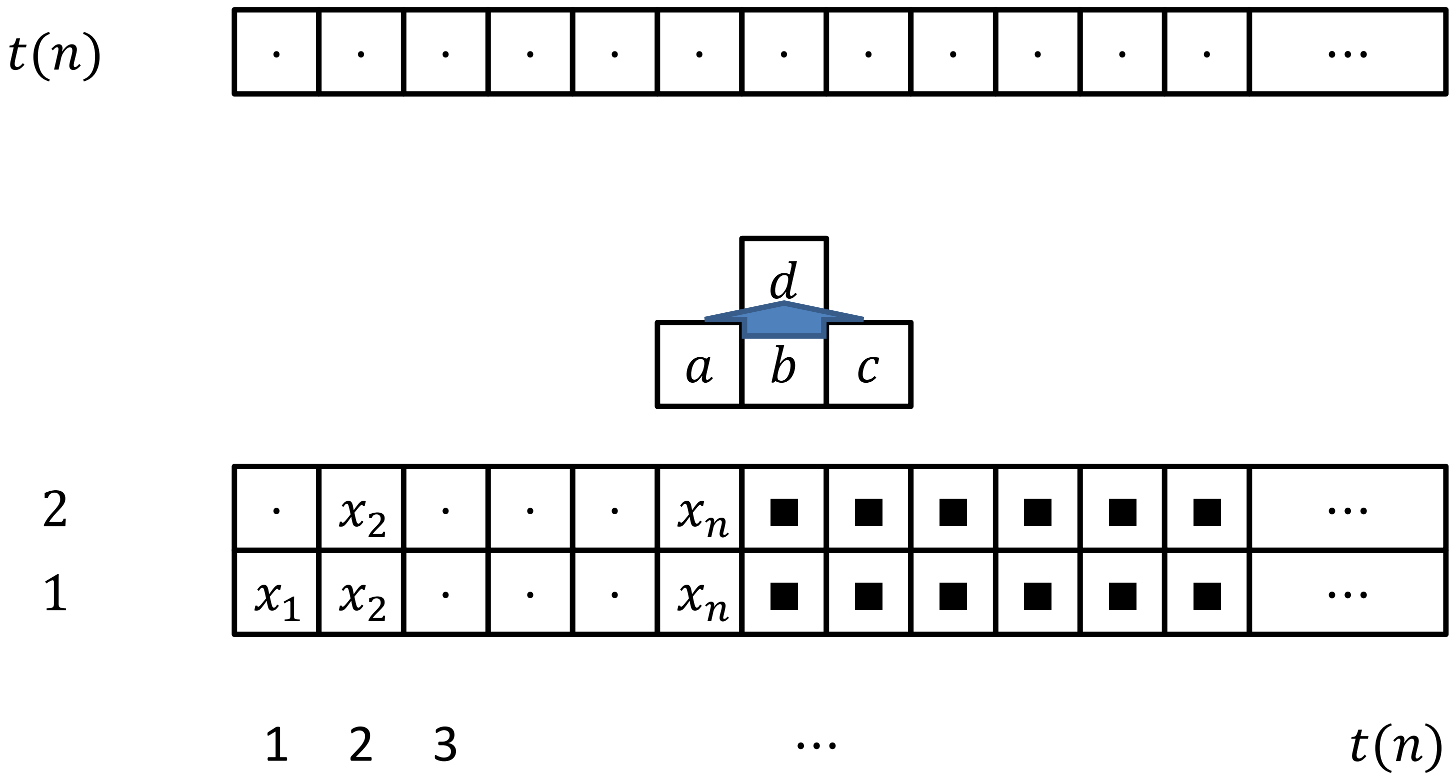


Turing machines into circuits

Pippenger-Fischer'77: DTIME($t(n)$) $\subseteq$ SIZE-DEPTH[$t(n) \log t(n)$, $t(n)$]

Turing machines into circuits

Exc: DTIME($t(n)$) $\subseteq$ SIZE-DEPTH$[t(n)^2, t(n)]$

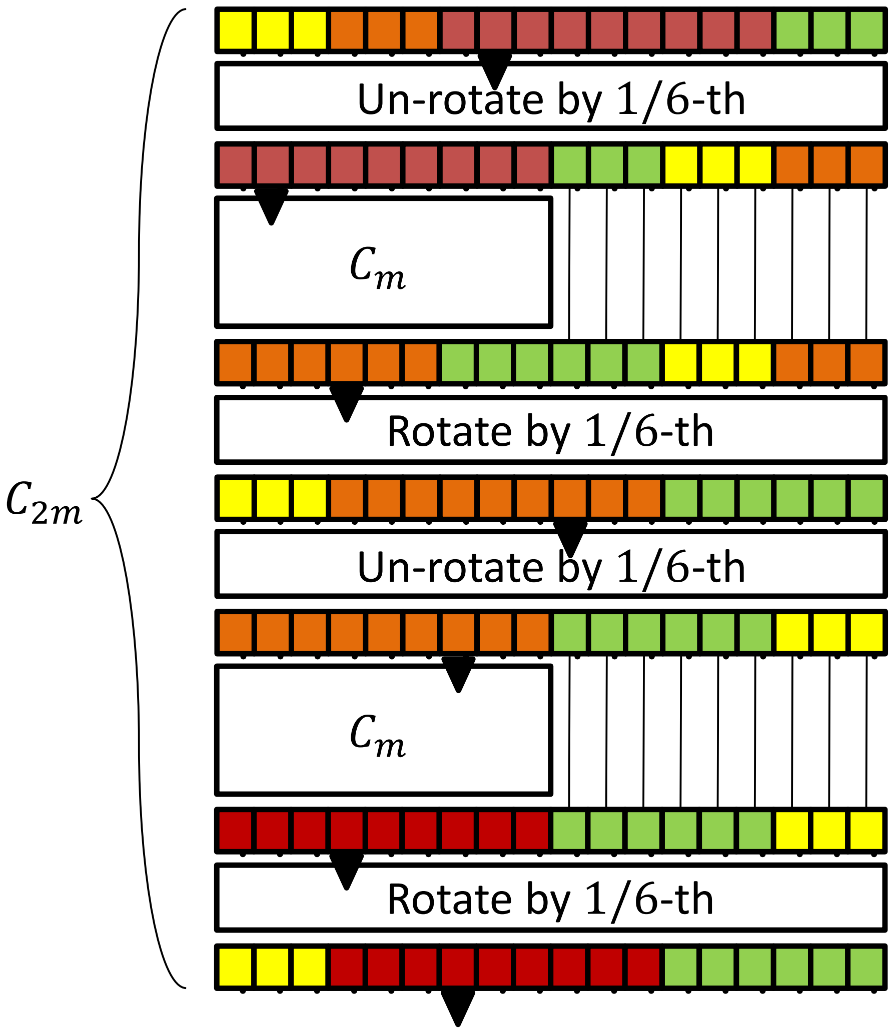


Turing machines into circuits

Pippenger-Fischer'77:
DTIME$(t(n)) \subseteq$
SIZE-DEPTH$[t(n) \log t(n), t(n)]$

$|C_{2m}| \leq 2\,|C_m| + O(m)$

$|C_m| \leq O(m \log m)$



Turing machines into circuits

Pippenger-Fischer'77: DTIME($t(n)$) ⊆ SIZE-DEPTH[$t(n) \log t(n)$, $t(n)$]

Unbounded fan-in gates:

DTIME($t(n)$) ⊆ SIZE-DEPTH[$t(n) \log t(n)$, $t(n)/\log\log t(n)$]

DTIME($t(n)$) ⊆ SIZE-DEPTH[$t(n)^{1+\varepsilon}$, $t(n)/\log n$]

Question: DTIME($t(n)$) ⊆ SIZE [$\mathrm{o}(t(n) \log t(n))$] ?

… for 1-tape TM ?

Sorting

Sorting: n integers, $2 \log n$ bits each.

TM	…	time $O(n \log^2 n)$
circuits	…	size $O(n \log^2 n)$

Question: Can you do better?

Turing machines into circuits