NL *and Parallel Computation*

TECHNISCHE UNIVERSITÄT DARMSTADT
Komplexitätstheorie

Every problem in **NL** can be solved in parallel time $O(\log^2 n)$ by *polynomially many processors*!

- dirPath $\leqslant_L$ Boolean Matrix powering:
  - $G=(V,E) \to$ adjacency matrix $A \in \{0,1\}^{V \times V}$ of G:
  - $A_{u,v}>0 \;:\Leftrightarrow\; v$ reachable from u in ≤1 step
  - $(A^k)_{u,v}>0 \Leftrightarrow v$ reachable from $u \in V$ in $\leq k$ steps
- goal: $(A^k)_{s,t}$ for some $k \geq |V| =: n$.
  - rept.squaring: $A \to A^2 \to A^4 \to A^8 \to \ldots$: $O(\log n)$
  - each phase = matrix multipl.; n^2 dot products
  - each dot product in parallel time $O(\log n)$

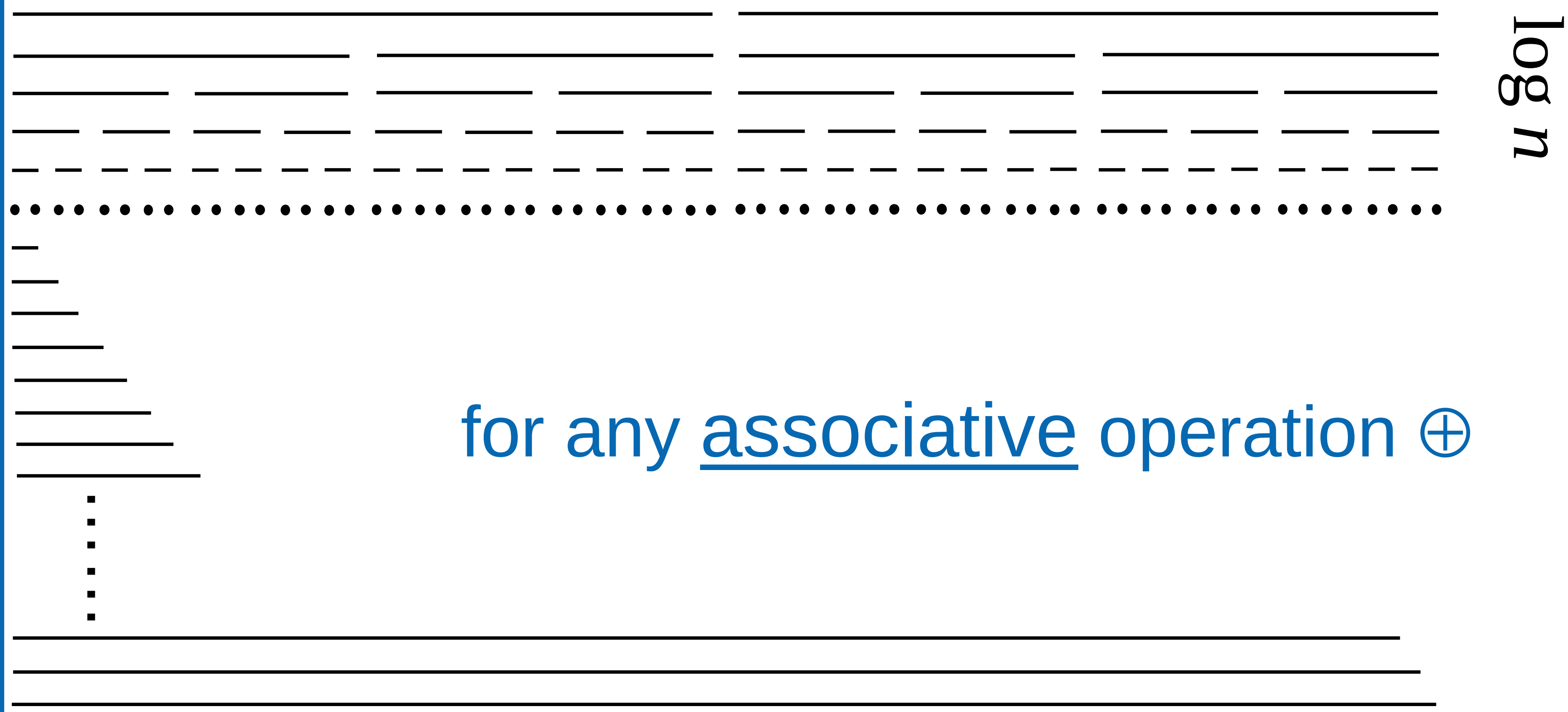
Theorem: $A \leqslant_L B \subseteq \Sigma^*$ solvable in parallel time $O(\log^k n)$ ($k \geq 2$) on polynomial size circuits $\Rightarrow$ same for A.

More Parallel Algorithms

Prefix Sum: Given $(x_1,\ldots,x_n)$, calculate <u>all</u> sums

$$x_1,\ x_1 \oplus x_2,\ x_1 \oplus x_2 \oplus x_3,\ \ldots,\ x_1 \oplus x_2 \oplus \ldots \oplus x_{n-1},\ x_1 \oplus x_2 \oplus \ldots \oplus x_{n-1} \oplus x_n$$

in logarithmic parallel time √ using $O(n \cdot \log n)$ gates



for any <u>associative</u> operation ⊕

Carry Look-Ahead Adder

Prefix Sum: Given $(x_1,\ldots,x_n)$, calculate all sums

$$x_1,\ x_1\oplus x_2,\ x_1\oplus x_2\oplus x_3,\ \ldots,\ x_1\oplus x_2\oplus\ldots\oplus x_{n-1},\ x_1\oplus x_2\oplus\ldots\oplus x_{n-1}\oplus x_n$$

in parallel time $O(\log n)$ for any associative operation $\oplus$

Long Addition: Given $(a_0,\ldots,a_{n-1})$ and $(b_0,\ldots,b_{n-1})$, calculate $(c_0,\ldots,c_{n-1},c_n) := (a_0,\ldots,a_{n-1}) + (b_0,\ldots,b_{n-1})$ in logarithmic parallel time? *ripple-carry adder*

i-th carry $z_i = g_i \vee (p_i \wedge z_{i-1})$

where $g_i := a_i \wedge b_i$

and $p_i := a_i \vee b_i$

'generate', 'propagate'

$$(z_i,0) = (z_{i-1},0) \otimes (g_i, p_i)$$

$$=((z_{i-2},0) \otimes (g_{i-1}, p_{i-1})) \otimes (g_i, p_i)$$

$$(g,p) \otimes (g',p') := (\ g'\vee(p'\wedge g),\ p'\wedge p\ )$$ associative!

Circuits: Depth and Size

TECHNISCHE UNIVERSITÄT DARMSTADT
Komplexitätstheorie

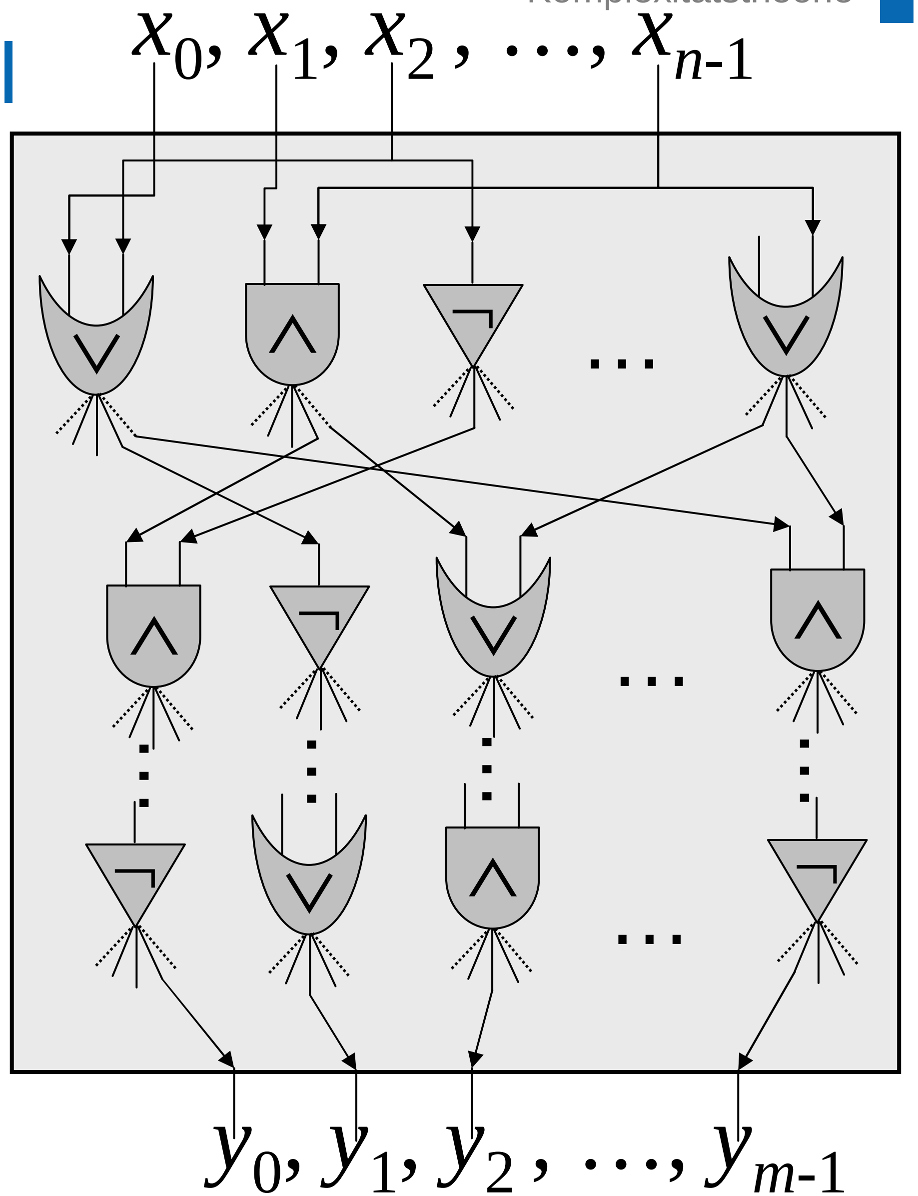
Gates ∨,∧,¬ are universal

unbounded fan-out

fan-in: binary/unary

N-ary: simulate
in depth O(log *N*)

- *n* inputs, *m* outputs
- depth *d* ⇒ size ≤ $m \cdot 2^d$
- If sorted topologically, evaluation on a TM in time O(size)



Uniformity

- Each circuit C has a fixed number of inputs

$\rightarrow$ for deciding $L \subseteq \{0,1\}^*$, consider a family (C_n)

- $\{1^n : n = \langle M \rangle$ for terminating TM $M\}$ undecidable to TM, but decidable by some family of circuits:
- F. Meyer auf der Heide (1984): knapsack can be decided by circuit family C_n of polynom.size
- New circuit for each n: nonuniform algorithm

Def: Call family C_n of circuits uniform if some logspace-DTM can, on input 1^n, output $\langle C_n \rangle$ (sorted topologically).

evaluation on a TM in time poly(size)

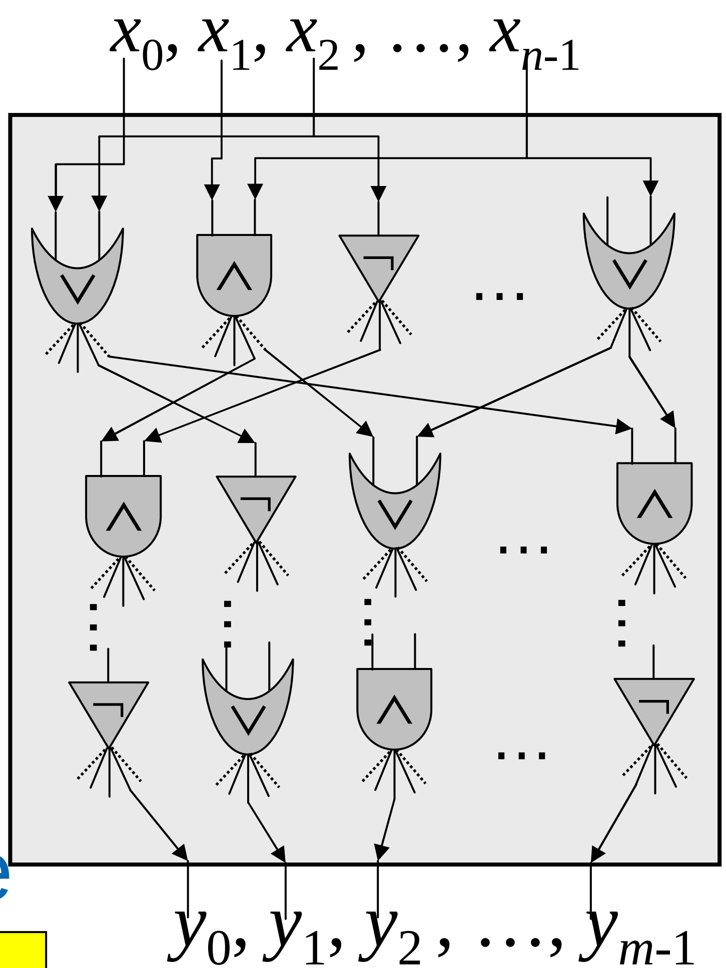
Circuit vs. Turing Complexity

Can evaluate a given circuit *C* on a TM

- in time O(size) once sorted topologically
- and in space O(*m*+depth):
  - for each gate on level *d*
  - recursively evaluate its 2 predecessors on levels<*d*

Can simulate a given TM *M* with input $\underline{x}$ on a circuit

- of depth $O(S_M(|\underline{x}|)^2)$
  Reachability + Matrix Powering
- of size $O(T_M(|\underline{x}|)^2)$: next slide



size ≈ seq. time, depth ≈ space

n inputs, *m* outputs