Lecture 19: Circuit Complexity

Real computers are built from gates.

Circuit complexity is a low-level model of computation.

Circuits are directed acyclic graphs. Inputs are placed at the leaves. Signals proceed up toward the root, r.

Straight-line code: gates are not reused.

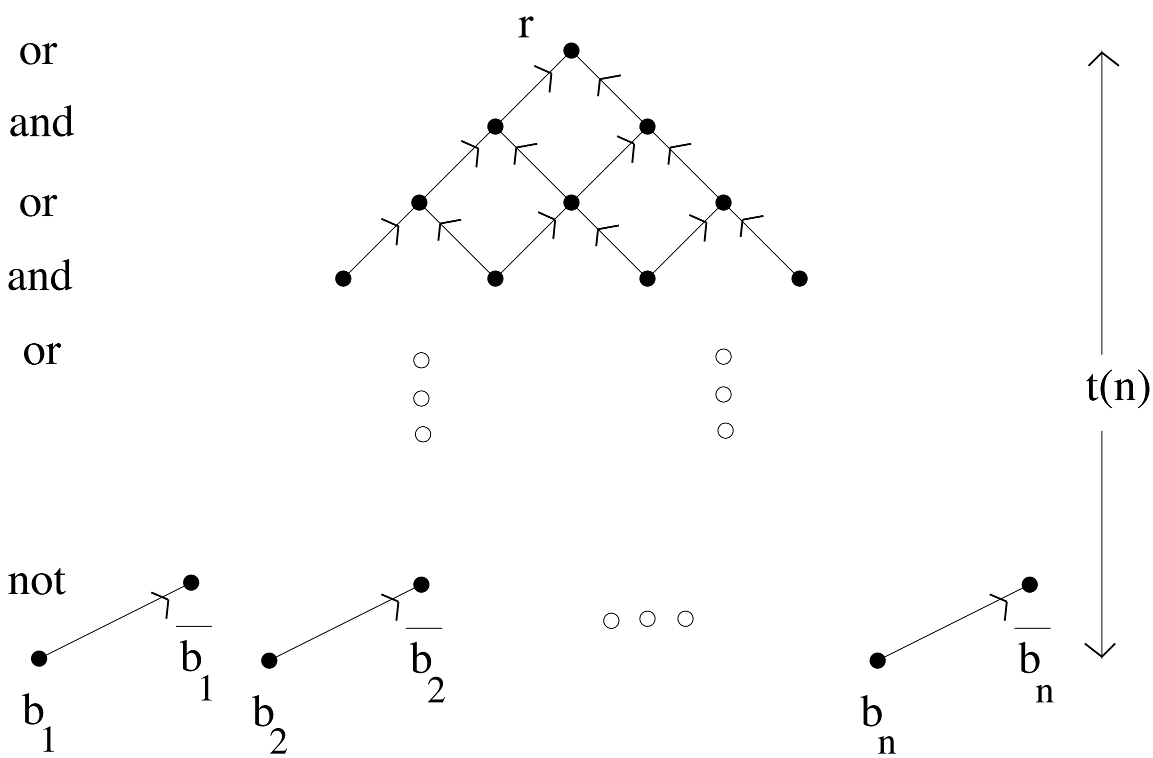
Let $S \subseteq \{0,1\}^\star$ be a decision problem.

Let, $C_1, C_2, C_3, \ldots$ be a circuit family.

C_n has n input bits and one output bit r.

Def: $\{C_i\}_{i \in \mathbf{N}}$ **computes** S iff for all n and for all $w \in \{0,1\}^n$,

$$w \in S \qquad \Leftrightarrow \qquad C_{|w|}(w) = 1 \; .$$



"not" gates are pushed down to bottom

Depth = parallel time

Number of gates = computational work = sequential time

Width = max number of gates at any level = amount of hardware in corresponding parallel machine

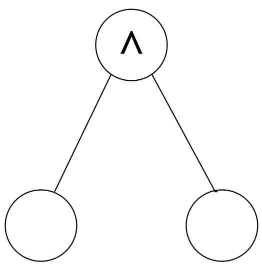
Circuit Complexity Classes

$S \subseteq \{0,1\}^\star$ is in NC$[t(n)]$, AC$t(n)$, ThC$t(n)$, iff exists uniform circuit family, $C_1, C_2, \ldots,$ s.t.

1. For all $w \in \{0,1\}^\star$, $\quad w \in S \quad \Leftrightarrow \quad C_{|w|}(w) = 1$
2. depth$(C_n) = O(t(n))$; $\quad |C_n| \leq n^{O(1)}$
3. The gates of C_n consist of,

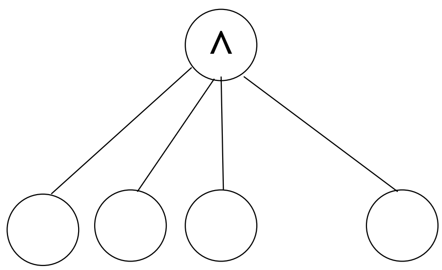
NC

bounded fan–in
and, or gates



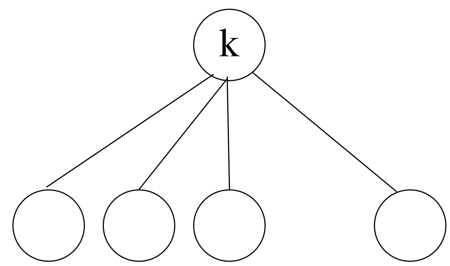
AC

unbounded fan–in
and, or gates



ThC

unbounded fan–in
threshold gates



Notation: for $i = 0, 1, \ldots,$ $\quad \text{NC}^i \; = \; \text{NC}[(\log n)^i];$

$$\text{AC}^i \; = \; \text{AC}(\log n)^i; \qquad \text{ThC}^i \; = \; \text{ThC}(\log n)^i$$

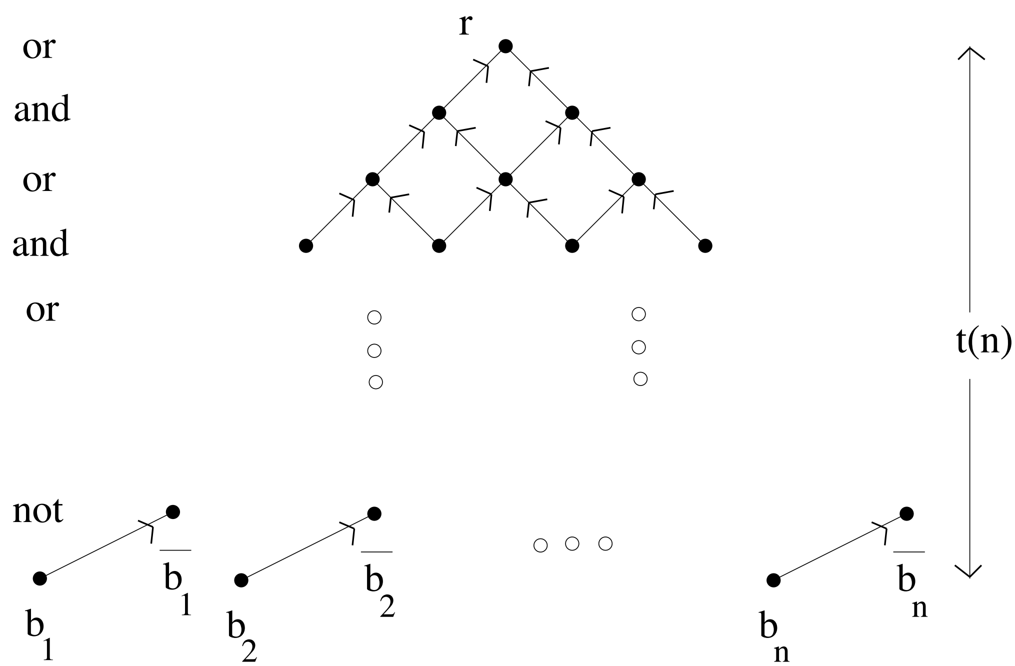
We will see that the following inclusions hold:

$$\begin{array}{ccccccccccc}
\text{AC}^0 & \subseteq & \text{ThC}^0 & \subseteq & \text{NC}^1 & \subseteq & \text{L} & \subseteq & \text{NL} & \subseteq & \text{AC}^1 \\
\text{AC}^1 & \subseteq & \text{ThC}^1 & \subseteq & \text{NC}^2 & & & & & \subseteq & \text{AC}^2 \\
\text{AC}^2 & \subseteq & \text{ThC}^2 & \subseteq & \text{NC}^3 & & & & & \subseteq & \text{AC}^3 \\
\vdots & \subseteq & \vdots & \subseteq & \vdots & & & & & \subseteq & \vdots
\end{array}$$

Thus:

$$\text{NC} \; = \; \bigcup_{i=0}^{\infty} \text{NC}^i \; = \; \bigcup_{i=0}^{\infty} \text{AC}^i \; = \; \bigcup_{i=0}^{\infty} \text{ThC}^i$$

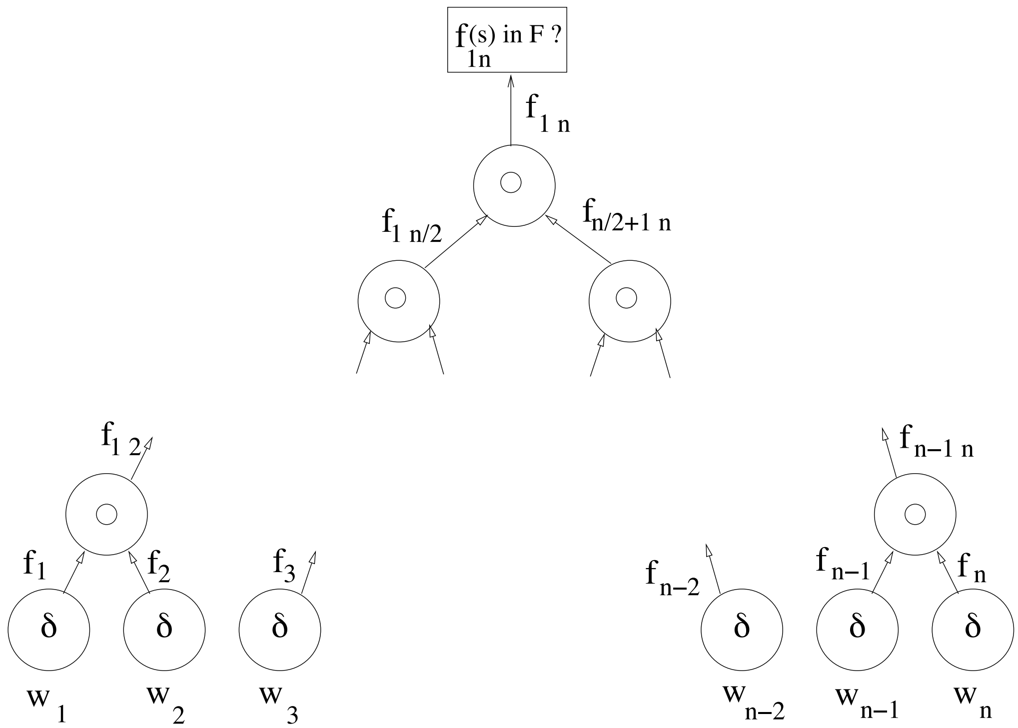
Uniform means that the map, $f : 1^n \mapsto C_n$ is **very easy**. $f \in F(\mathrm{L})$; $f \in F(\mathrm{FO})$



Each C_i is an instance of the same program.

Prop: Every regular language is in NC^1.

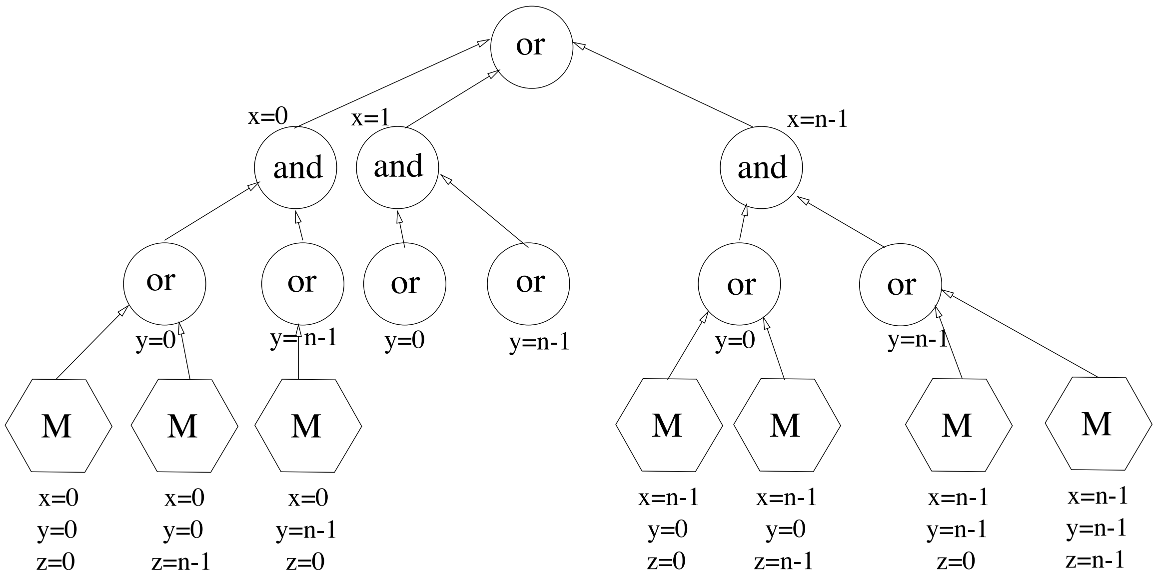
Proof: DFA $D = \langle \Sigma, Q, \delta, s, F \rangle$. Build circuits: $C_1, C_2, \ldots,$



$f_i(q) \;=\; \delta(q, w_i);$ $\qquad w \in \mathcal{L}(D) \quad \Leftrightarrow \quad f_{1n}(s) \in F$ $\qquad \square$

Thm: FO $=$ AC^0

Example: $$\varphi \equiv \exists x \, \forall y \, \exists z \, (M(x,y,z))$$



Prop: For $i = 0, 1, \ldots,$

$$\mathrm{NC}^i \quad \subseteq \quad \mathrm{AC}^i \quad \subseteq \quad \mathrm{ThC}^i \quad \subseteq \quad \mathrm{NC}^{i+1}$$

Proof: All inclusions except $\mathrm{ThC}^i \subseteq \mathrm{NC}^{i+1}$ are clear.

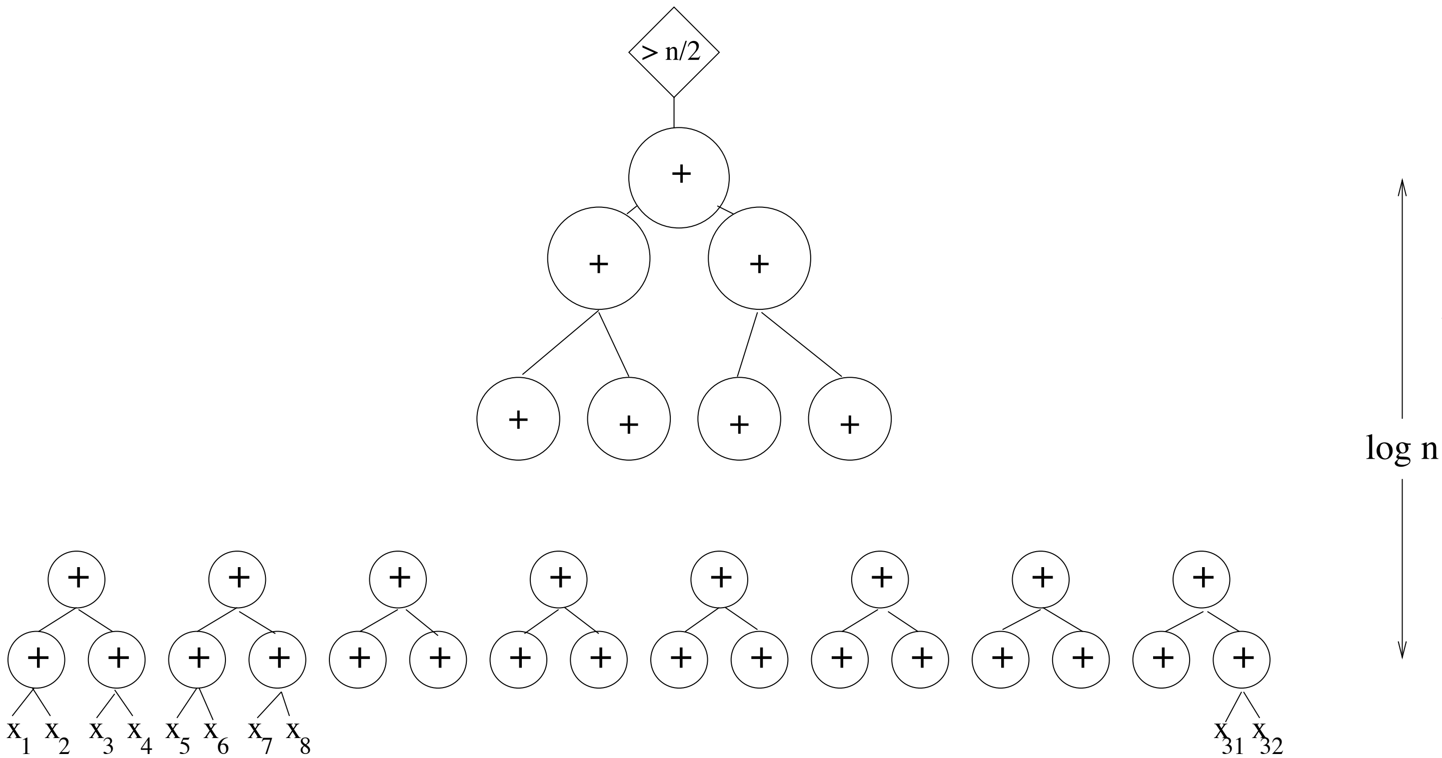
$$\mathrm{MAJ} \;=\; \big\{ w \in \{0,1\}^\star \;\big|\; w \text{ has more than } |w|/2 \text{ "1"s} \big\} \;\in \mathrm{ThC}^0$$

Lemma: $\mathrm{MAJ} \in \mathrm{NC}^1$

(and the same for any other threshold gate).

Try to build an NC[1] circuit for majority by adding the n input bits via a full binary tree of height $\log n$.

Problem: the sums being added have more and more bits; still want to add them in constant depth.



Solution: Ambiguous Notation

Binary representation; but with digits: $0, 1, 2, 3$

$$\begin{aligned} 3213 &= 3 \cdot 2^3 + 2 \cdot 2^2 + 1 \cdot 2^1 + 3 \cdot 2^0 &= 37 \\ 3221 &= 3 \cdot 2^3 + 2 \cdot 2^2 + 2 \cdot 2^1 + 1 \cdot 2^0 &= 37 \end{aligned}$$

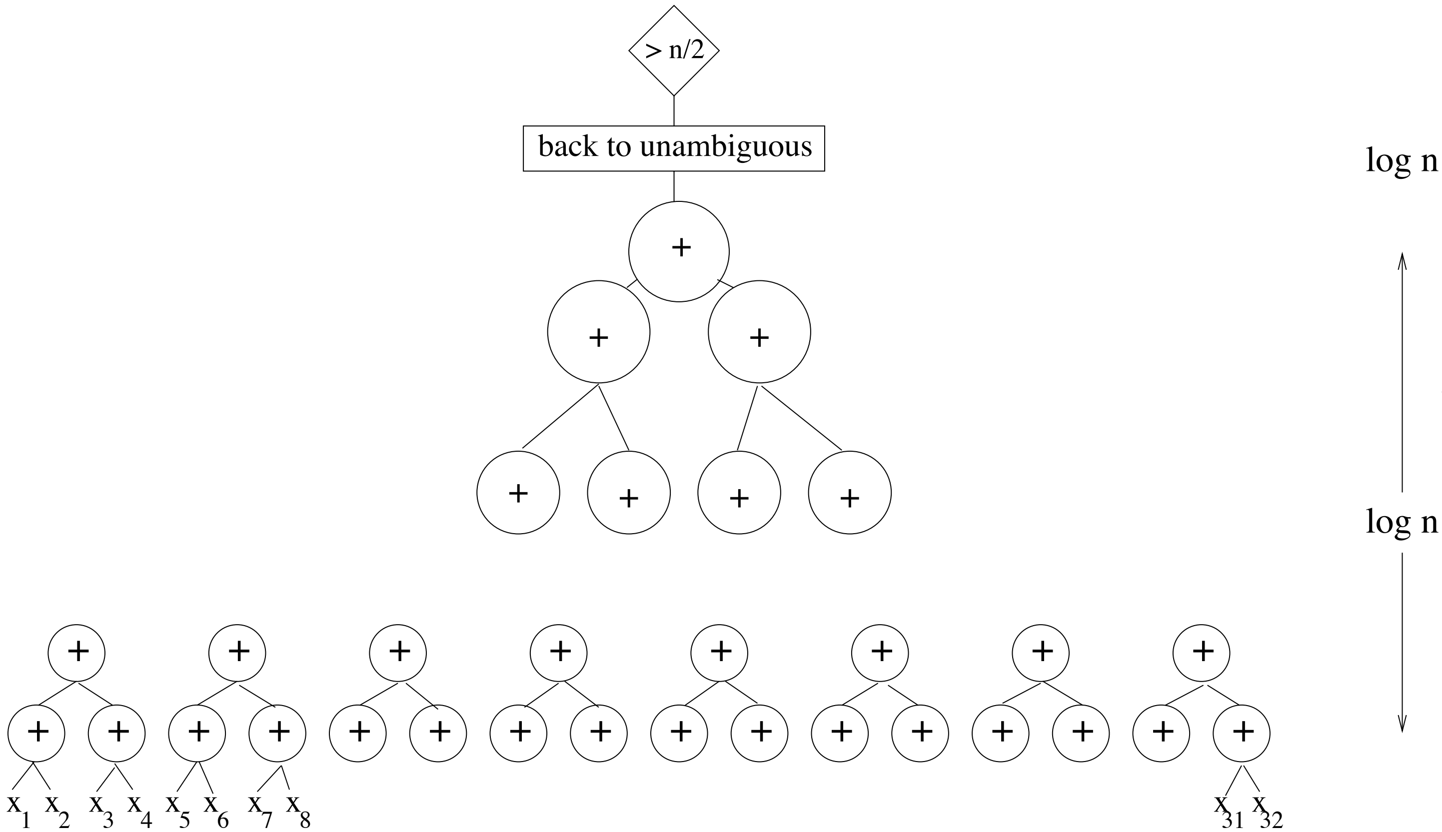
Lemma: Ambiguous Notation Addition is in NC^0

Example:

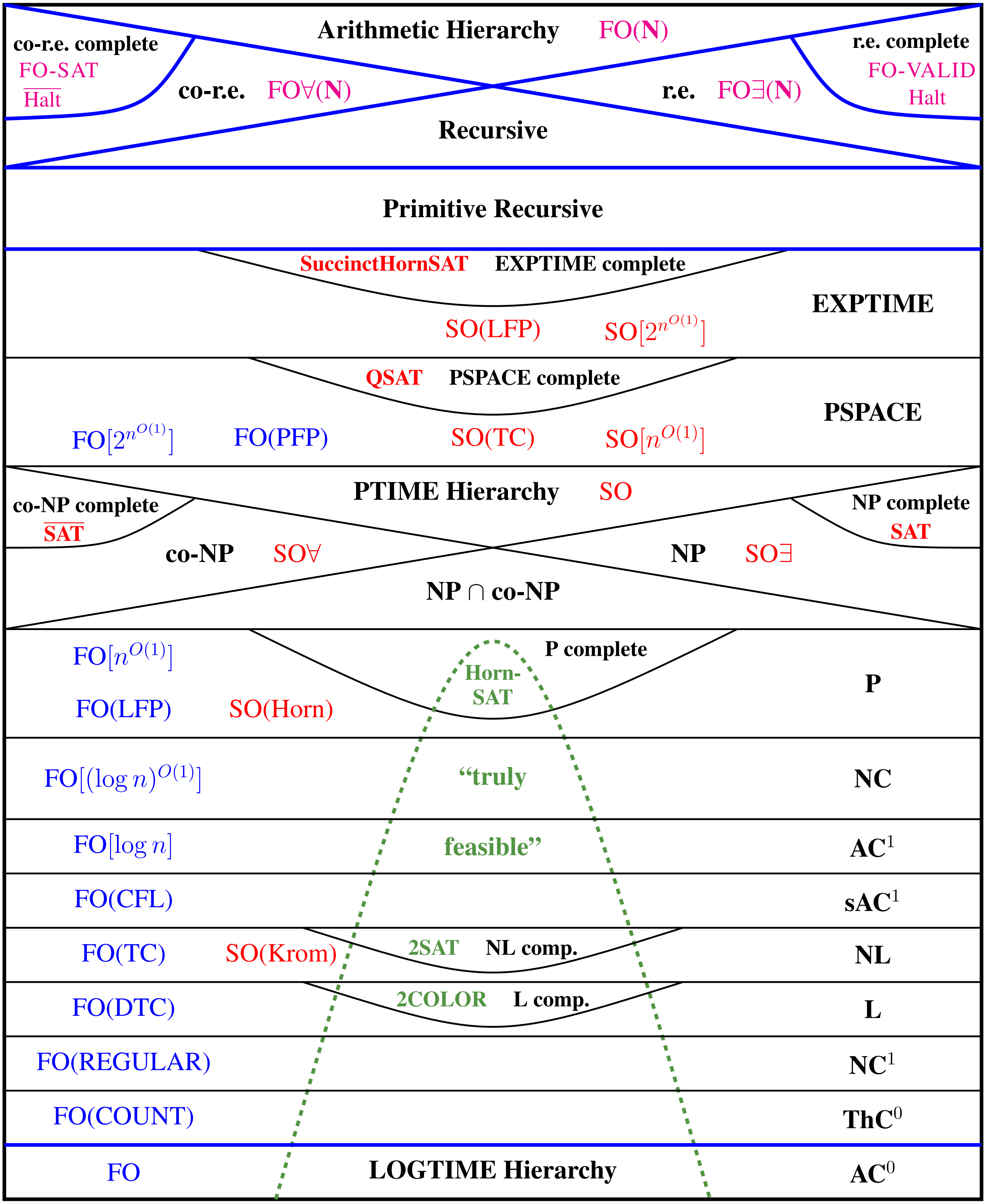
$$\begin{array}{lccccc} \text{carries:} & 3 & 2 & 2 & 3 & \\ & & 3 & 2 & 1 & 3 \\ + & & 3 & 2 & 1 & 3 \\ \hline & 3 & 2 & 2 & 1 & 0 \end{array}$$

The carry from column i is determined by columns i and $i+1$: use the largest carry we are sure to get.

Translating from ambiguous to binary, is just addition, thus first-order, thus AC^0, and thus NC^1.



□



Arithmetic Hierarchy
FO(**N**)
co-r.e. complete
FO-SAT
$\overline{\text{Halt}}$
co-r.e.
FO∀(**N**)
r.e.
FO∃(**N**)
r.e. complete
FO-VALID
Halt
Recursive
Primitive Recursive
SuccinctHornSAT
EXPTIME complete
EXPTIME
SO(LFP)
$\mathrm{SO}[2^{n^{O(1)}}]$
QSAT
PSPACE complete
PSPACE
$\mathrm{FO}[2^{n^{O(1)}}]$
FO(PFP)
SO(TC)
$\mathrm{SO}[n^{O(1)}]$
PTIME Hierarchy
SO
co-NP complete
$\overline{\text{SAT}}$
co-NP
SO∀
NP
SO∃
NP complete
SAT
NP ∩ co-NP
$\mathrm{FO}[n^{O(1)}]$
P complete
Horn-SAT
P
FO(LFP)
SO(Horn)
$\mathrm{FO}[(\log n)^{O(1)}]$
“truly
NC
$\mathrm{FO}[\log n]$
feasible”
AC^1
FO(CFL)
sAC^1
FO(TC)
SO(Krom)
2SAT
NL comp.
NL
FO(DTC)
2COLOR
L comp.
L
FO(REGULAR)
NC^1
FO(COUNT)
ThC^0
FO
LOGTIME Hierarchy
AC^0