Time and Space functions: $t, s : \mathbf{N} \to \mathbf{N}^+$

Definition 5.1 A set $A \subseteq U$ is in DTIME$[t(n)]$ iff there exists a deterministic, multi-tape TM, M, and a constant c, such that,

1. $A \quad = \quad \mathcal{L}(M) \quad \equiv \quad \{w \in U \mid M(w) = 1\}$, and
2. $\forall w \in U$, $M(w)$ halts within $c \cdot t(|w|)$ steps. □

Definition 5.2 A set $A \subseteq U$ is in DSPACE$[s(n)]$ iff there exists a deterministic, multi-tape TM, M, and a constant c, such that,

1. $A \quad = \quad \mathcal{L}(M)$, and
2. $\forall w \in U$, $M(w)$ uses at most $c \cdot s(|w|)$ work-tape cells.

(Input tape is "read-only" and not counted as space used.) □

Example: PALINDROMES $\in$ DTIME$[n]$, DSPACE$[n]$.

In fact, PALINDROMES $\in$ DSPACE$[\log n]$. [Exercise]

Definition 5.3 $f : U \to U$ is in $F(\text{DTIME}[t(n)])$ iff there exists a deterministic, multi-tape TM, M, and a constant c, such that,

1. $f \quad = \quad M(\cdot)$;
2. $\forall w \in U$, $M(w)$ halts within $c \cdot t(|w|)$ steps;
3. $|f(w)| \leq |w|^{O(1)}$, i.e., f is polynomially bounded.

$\square$

Definition 5.4 $f : U \to U$ is in $F(\text{DSPACE}[s(n)])$ iff there exists a deterministic, multi-tape TM, M, and a constant c, such that,

1. $f \quad = \quad M(\cdot)$;
2. $\forall w \in U$, $M(w)$ uses at most $c \cdot s(|w|)$ work-tape cells;
3. $|f(w)| \leq |w|^{O(1)}$, i.e., f is polynomially bounded.

(Input tape is "read-only"; Output tape is "write-only". Neither is counted as space used.) $\square$

Example: Plus $\in F(\text{DTIME}[n])$, Mult $\in F(\text{DTIME}[n^2])$

$$\begin{aligned} \text{L} &\equiv \text{DSPACE}[\log n] \\ \text{P} &\equiv \text{DTIME}[n^{O(1)}] \equiv \bigcup_{i=1}^{\infty} \text{DTIME}[n^i] \\ \text{PSPACE} &\equiv \text{DSPACE}[n^{O(1)}] \equiv \bigcup_{i=1}^{\infty} \text{DSPACE}[n^i] \end{aligned}$$

These classes will become your good friends soon.

Theorem 5.5 *For any functions $t(n) \geq n,\ s(n) \geq \log n$, we have*

$$\begin{aligned} \mathrm{DTIME}[t(n)] &\subseteq \mathrm{DSPACE}[t(n)] \\ \mathrm{DSPACE}[s(n)] &\subseteq \mathrm{DTIME}[2^{O(s(n))}] \end{aligned}$$

Proof: Let M be a DSPACE$[s(n)]$ TM, let $w \in \Sigma_0^\star$, let $n = |w|$

$M(w)$ has k tapes and uses at most $cs(n)$ work-tape cells.

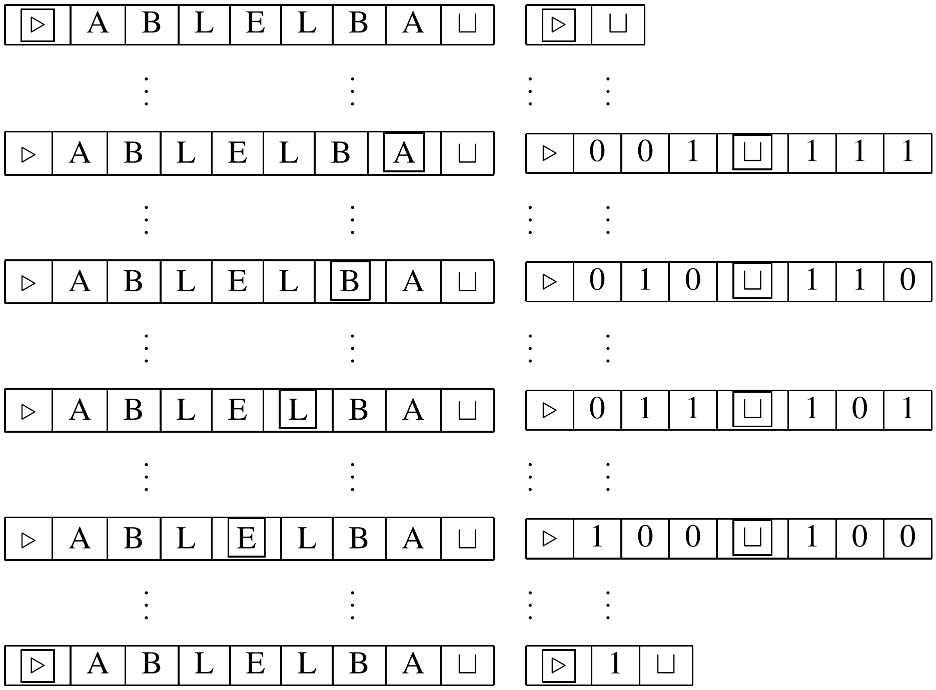
$M(w)$ has at most $2^{k's(n)}$ possible configurations:

$$|Q| \quad \cdot \quad (n + cs(n) + 2)^k \quad \cdot \quad |\Sigma|^{cs(n)} \quad < \quad 2^{k's(n)}$$

# of states · # of head positions · # of tape contents

Thus, after $2^{k's(n)}$ steps, $M(w)$ must be in an infinite loop. □

Corollary 5.6 L $\subseteq$ P $\subseteq$ PSPACE



Using $O(\log n)$ workspace, we can keep track of and manipulate two pointers into the input.

RAM = Random Access Machine

Memory:

κ	r_0	r_1	r_2	r_3	r_4	$\cdots$	r_i	$\cdots$

κ = program counter; r_0 = accumulator

Instruction	**Operand**	**Semantics**
READ	$j \mid \uparrow j \mid = j$	$r_0 := (r_j \mid r_{r_j} \mid j)$
STORE	$j \mid \uparrow j$	$(r_j \mid r_{r_j}) := r_0$
ADD	$j \mid \uparrow j \mid = j$	$r_0 := r_0 + (r_j \mid r_{r_j} \mid j)$
SUB	$j \mid \uparrow j \mid = j$	$r_0 := r_0 - (r_j \mid r_{r_j} \mid j)$
HALF		$r_0 := \lfloor r_0/2 \rfloor$
JUMP	j	$\kappa := j$
JPOS	j	**if** $(r_0 > 0)$ **then** $\kappa := j$
JZERO	j	**if** $(r_0 = 0)$ **then** $\kappa := j$
HALT		$\kappa := 0$

Theorem 5.7 DTIME$[t(n)]$ $\subseteq$ RAM-TIME$[t(n)]$ $\subseteq$ DTIME$[(t(n))^3]$

Proof: Memorize program in finite control. Store all registers on one tape:

$\triangleright$	1	1	,	0	:	1	0	1	,	1	0	1	:	0	,	1	0	1	1	:	1	0	$\sqcup$
	κ					r_0						r_5							r_{11}				

Store workspace for calculations on second tape:

$\triangleright$	1	0	0	,	1	0	1	1	$\sqcup$
		κ'					A		

Use the third tape for copying and pasting sections of the first tape.

$\triangleright$	0	:	1	0	1	,	1	0	1	:	0	,	1	0	1	1	:	1	0	$\sqcup$
			r_0						r_5							r_{11}				

Each register contains at most $n + t(n)$ bits. [$O(\log n)$ would be more realistic.]

The total number of tape cells used is at most $2t(n)(n + t(n)) = O((t(n))^2)$.

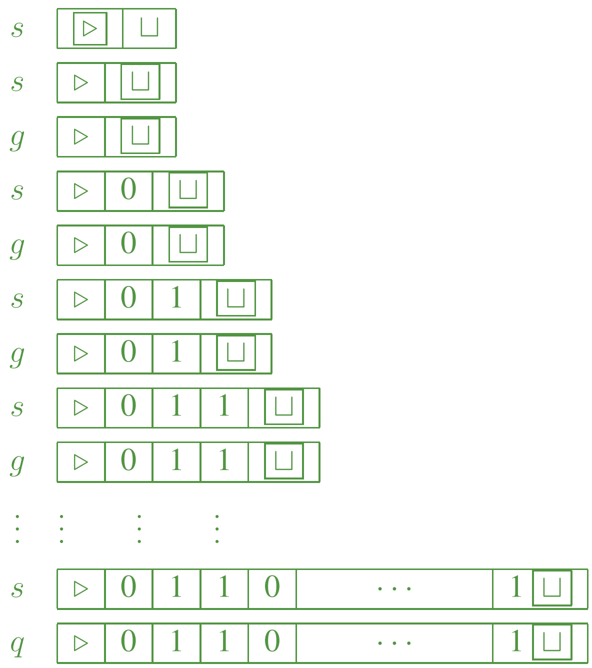
Each step takes at most $O((t(n))^2)$ steps to simulate. $\square$

Nondeterministic Turing Machines choose one of two possible moves each step.

guess.tm	s	g	q
0			
1			
$\sqcup$	$g, \sqcup, - \mid q, \sqcup, -$	$s, 0, \rightarrow \mid s, 1, \rightarrow$	
$\triangleright$	$s, \triangleright, \rightarrow$		
comment	g or q	guess 0 or 1	the rest

Nondeterministic Guess Machine is a typical example:

- Write down an arbitrary string, $g \in \{0, 1\}^\star$: the guess.
- Proceed with the rest of the computation, using g if desired.
- Accept iff there exists some guess that leads to acceptance.



guess.tm	s	g	q
0			
1			
$\sqcup$	$g, \sqcup, -$ | $q, \sqcup, -$	$s, 0, \rightarrow$ | $s, 1, \rightarrow$	
$\triangleright$	$s, \triangleright, \rightarrow$		
comment	g or q	guess 0 or 1	the rest

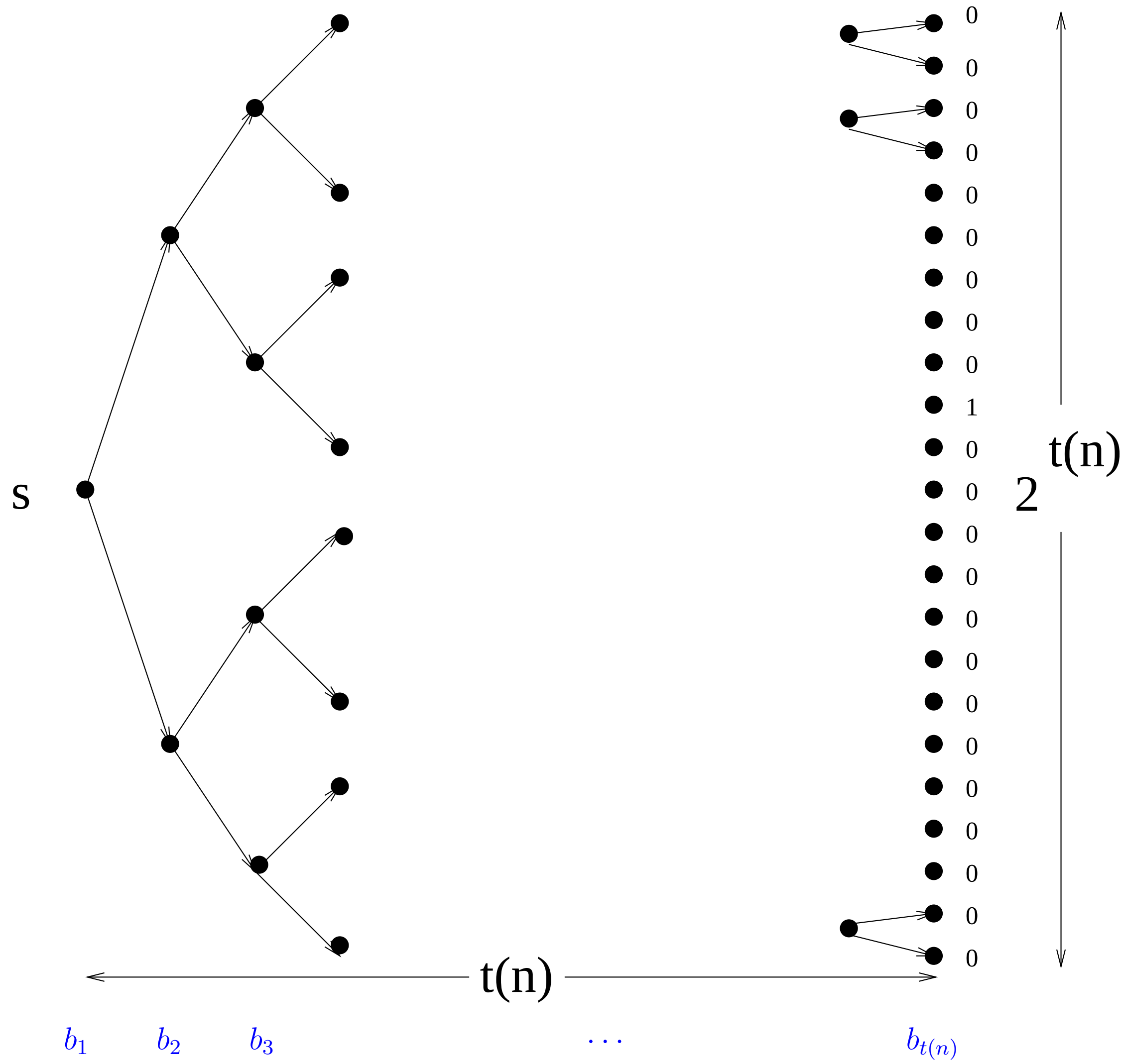
Definition 5.8 The **set** accepted by a NTM, N:

$$\mathcal{L}(N) \quad \equiv \quad \left\{ w \in U \;\middle|\; \text{some run of } N(w) \text{ halts with output "1"} \right\}$$

The **time** taken by N on $w \in \mathcal{L}(N)$ is the **number of steps** in the **shortest computation** of $N(w)$ that accepts. □

Unfortunately, this is a mathematical fiction.

As far as we know, you can't **really** build a nondeterministic Turing Machine.


s
0
0
0
0
0
0
0
0
0
1
0
0
0
0
0
0
0
0
0
0
0
0
0
$2^{t(n)}$
t(n)
b_1
b_2
b_3
$\cdots$
$b_{t(n)}$

$$\text{NTIME}[t(n)] \equiv \text{problems accepted by NTMs in time } O(t(n))$$

$$\text{NP} \equiv \text{NTIME}[n^{O(1)}] \equiv \bigcup_{i=1}^{\infty} \text{NTIME}[n^i]$$

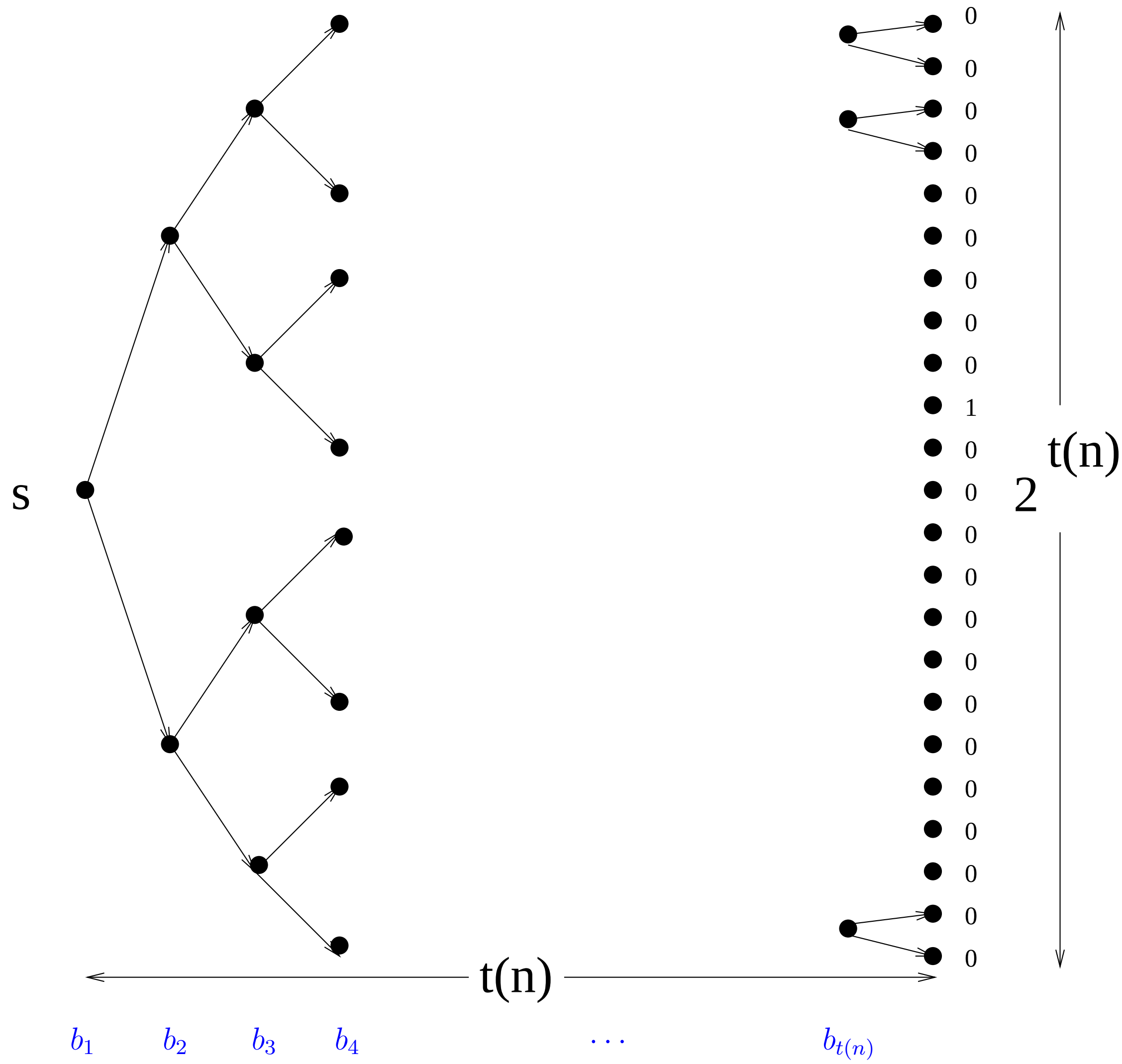
Theorem 5.9 *For any function $t(n)$,*

$$\text{DTIME}[t(n)] \subseteq \text{NTIME}[t(n)] \subseteq \text{DSPACE}[t(n)] \subseteq \text{DTIME}[2^{O(t(n))}]$$

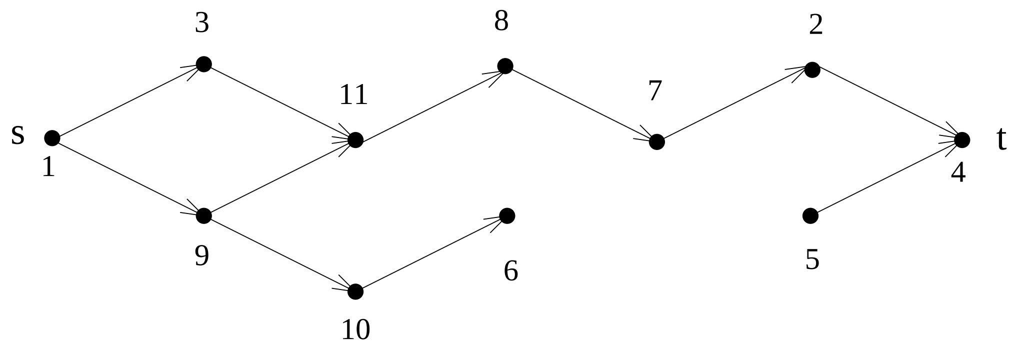
Corollary 5.10 L $\subseteq$ P $\subseteq$ NP $\subseteq$ PSPACE

Corollary 5.11 *The definition of* **Recursive** *and* **r.e.** *are unchanged if we use nondeterministic instead of deterministic Turing machines.*

NSPACE$[s(n)]$ is the set of problems accepted by NTMs using at most $O(s(n))$ space on each branch. [Can run in time $t(n) \leq 2^{O(s(n))}$.]

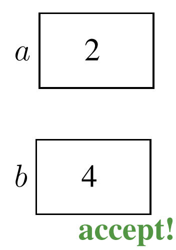


Definition 5.12 REACH $= \{G \mid s \xrightarrow{\star} t\}$ $\square$



Prop: REACH $\in$ NL $=$ NSPACE$[\log n]$

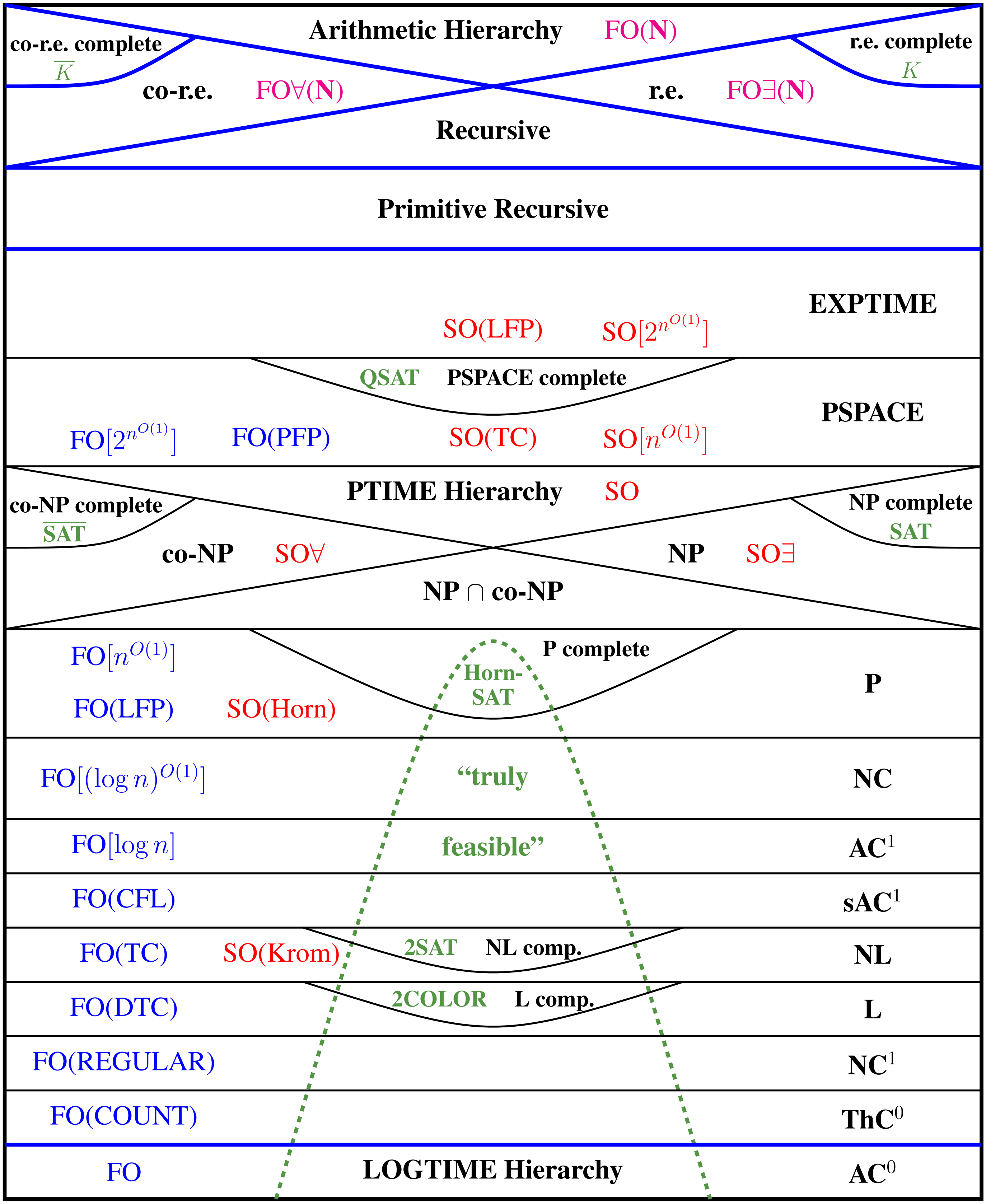
$b := s$
for $c := 1$ **to** $n = |V|$ **do** $\{$
  if $b = t$ **then accept**
  $a := b$
  choose new b
  if $(\neg E(a,b))$ **then reject** $\}$
reject



Def: Problem T is **complete** for complexity class **C** iff

1. $T \in \mathbf{C}$, and
2. $\forall A \in \mathbf{C}\,(A \leq T)$

Reductions now must be in $F(\mathrm{L})$.


Arithmetic Hierarchy
FO(N)
co-r.e. complete
$\overline{K}$
co-r.e.
FO∀(N)
r.e. complete
K
r.e.
FO∃(N)
Recursive
Primitive Recursive
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
$\overline{\mathrm{SAT}}$
co-NP
SO∀
NP complete
SAT
NP
SO∃
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