Turing Machines: $M = (Q, \Sigma, \delta, s)$

$$\delta : Q \times \Sigma \rightarrow (Q \cup \{h\}) \times \Sigma \times \{\leftarrow, \rightarrow, -\}$$

Def: Function f is *recursive* iff it is computed by a TM. f may be total or partial.

Def: A set S is *recursive* iff its characteristic function χ_S is a recursive function.

Recursive is the set of recursive sets.

A set S is *recursively enumberable (r.e.)* iff its partial characteristic function p_S is a recursive function.

r.e. is the set of r.e. sets.

Th: **Recursive** = **r.e.** $\cap$ **co-r.e.**

Definition 5.1 A string $w \in \Sigma^\star$ is a *palindrome* iff it is the same as its reversal, i.e., $w = w^R$. ♠

Examples of palindromes:

- 101
- 1101001011
- ABLE WAS I ERE I SAW ELBA
- AMANAPLANACANALPANAMA

Fact 5.2 *The set of PALINDROMES (over a fixed alphabet, Σ is context-free but not regular.*

Proposition 5.3 *The set of PALINDROMES (over a fixed alphabet, Σ) is a recursive set.*

Proof:

$\boxed{\triangleright}$	A	B	L	E	E	L	B	A	$\sqcup$

♠

Fact 5.4 *Time $O(n^2)$ is necessary and sufficient for a one-tape Turing machine to accept the set, PALINDROMES.*

Proof: Time $O(n^2)$ suffices. One way to see this is to do problems 2.8.4, 2.8.5 from [P]. ♠

Definition 5.5 A k-tape Turing machine, $M = (Q, \Sigma, \delta, s)$

Q: finite set of states; $s \in Q$

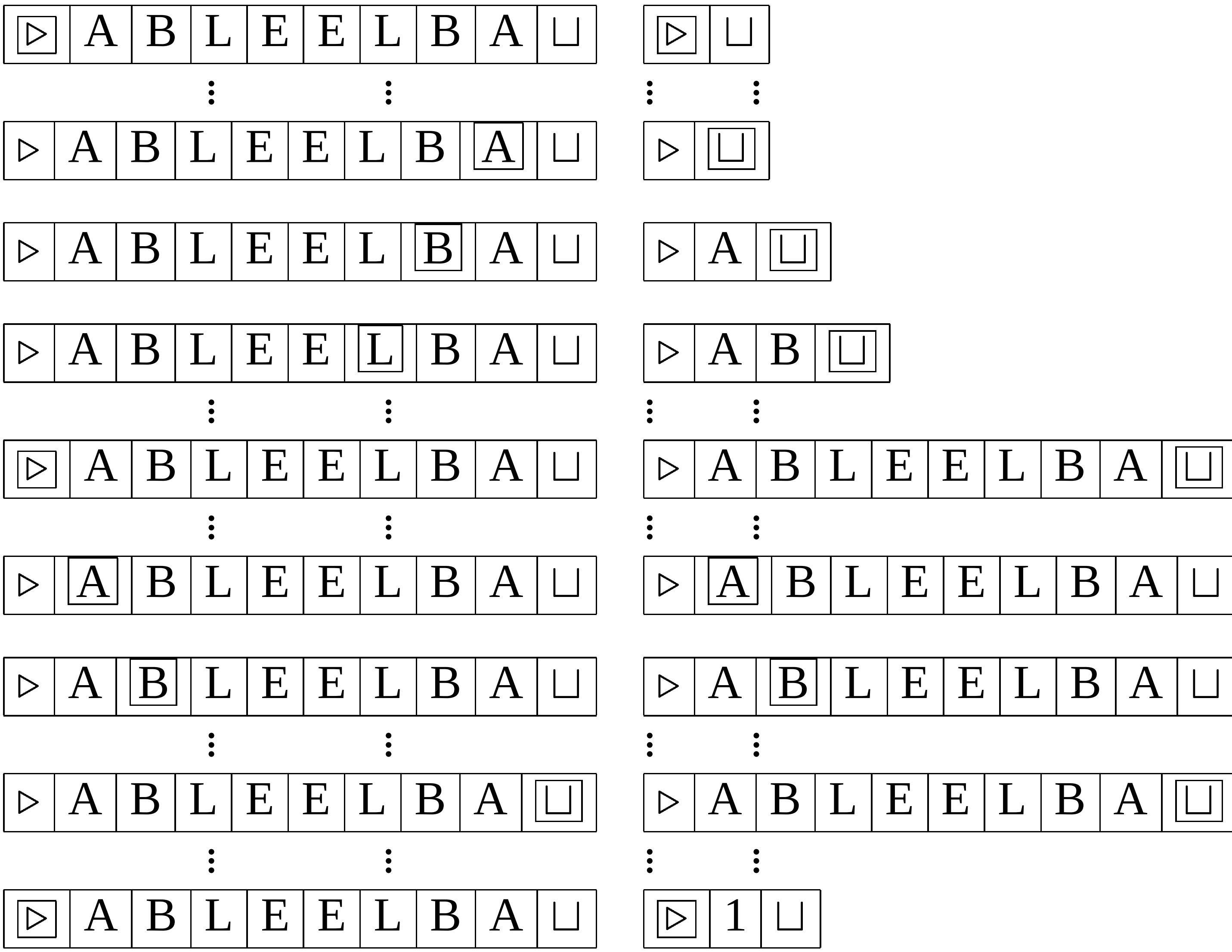
Σ: finite set of symbols;

δ:$Q \times \Sigma^k \ \rightarrow \ (Q \cup \{h\}) \times (\Sigma \times \{\leftarrow, \rightarrow, -\})^k$

♠

Proposition 5.6 *PALINDROMES can be accepted in* **DTIME**$[n]$ *on a 2-tape TM.*

Proof: (that PALINDROMES $\in$ **DTIME**$[n]$)



♠

Definition 5.7 A set $A \subseteq \Sigma^\star$ is in **DTIME**$[t(n)]$ iff there exists a deterministic, multi-tape TM, M, and a constant c, such that,

1. $A = \mathcal{L}(M) \equiv \{w \in \Sigma^\star \mid M(w) = 1\}$, and
2. $\forall w \in \Sigma^\star$, $M(w)$ halts within $c(1 + t(|w|))$ steps.

♠

Definition 5.8 A set $A \subseteq \Sigma^\star$ is in **DSPACE**$[s(n)]$ iff there exists a deterministic, multi-tape TM, M, and a constant c, such that,

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

(Note: The input tape is **read-only** and **not counted as space used**. Otherwise space bounds below n would rarely be useful. But in the real world we often want to limit space and work with read-only input. ♠

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

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

Definition 5.9 $f : \Sigma^\star \to \Sigma^\star$ is in $F(\textbf{DTIME}[t(n)])$ iff there exists a deterministic, multi-tape TM, M, and a constant c, such that,

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

♠

Definition 5.10 $f : \Sigma^\star \to \Sigma^\star$ is in $F(\textbf{DSPACE}[s(n)])$ iff there exists a deterministic, multi-tape TM, M, and a constant c, such that,

1. $f \quad = \quad M(\cdot)$;
2. $\forall w \in \Sigma^\star$, $M(w)$ uses at most $c(1 + s(|w|))$ 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.) ♠

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

$$\mathbf{L} \equiv \mathbf{DSPACE}[\log n]$$

$$\mathbf{P} \equiv \mathbf{DTIME}[n^{O(1)}] \equiv \bigcup_{i=1}^{\infty} \mathbf{DTIME}[n^i]$$

$$\mathbf{PSPACE} \equiv \mathbf{DSPACE}[n^{O(1)}] \equiv \bigcup_{i=1}^{\infty} \mathbf{DSPACE}[n^i]$$

Theorem 5.11 *For any functions $t(n) \geq n, s(n) \geq \log n$, we have*

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

Proof: Let M be a **DSPACE**$[s(n)]$ TM,

let $w \in \Sigma^\star$, let $n = |w|$

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

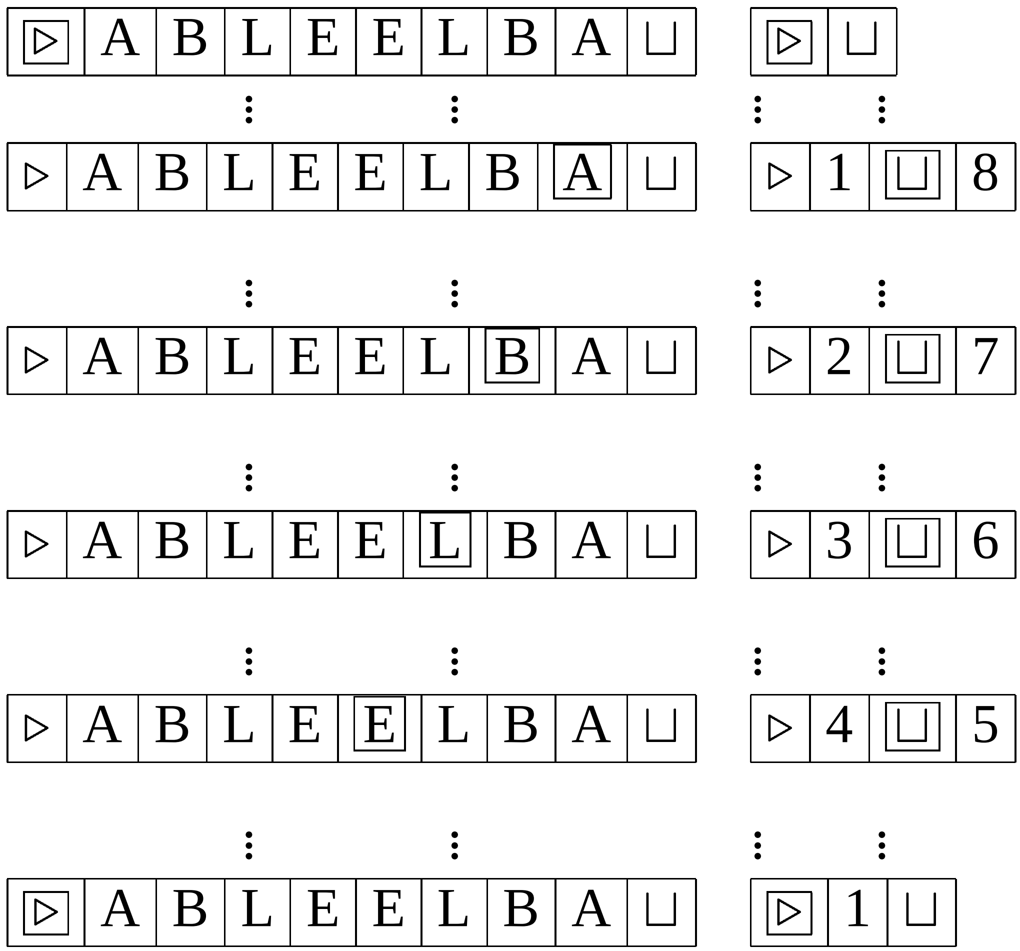
$$|Q| \cdot (n + cs(n) + 2)^k \cdot |\Sigma|^{cs(n)} \quad < \quad 2^{k' s(n)}$$

possible configurations.

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

Corollary 5.12 **L** $\subseteq$ **P** $\subseteq$ **PSPACE**

PALINDROMES $\in$ L



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 5.13

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 moving over 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.

The total number of tape cells used is at most

$$2t(n)(n + t(n)) = O((t(n))^2)$$

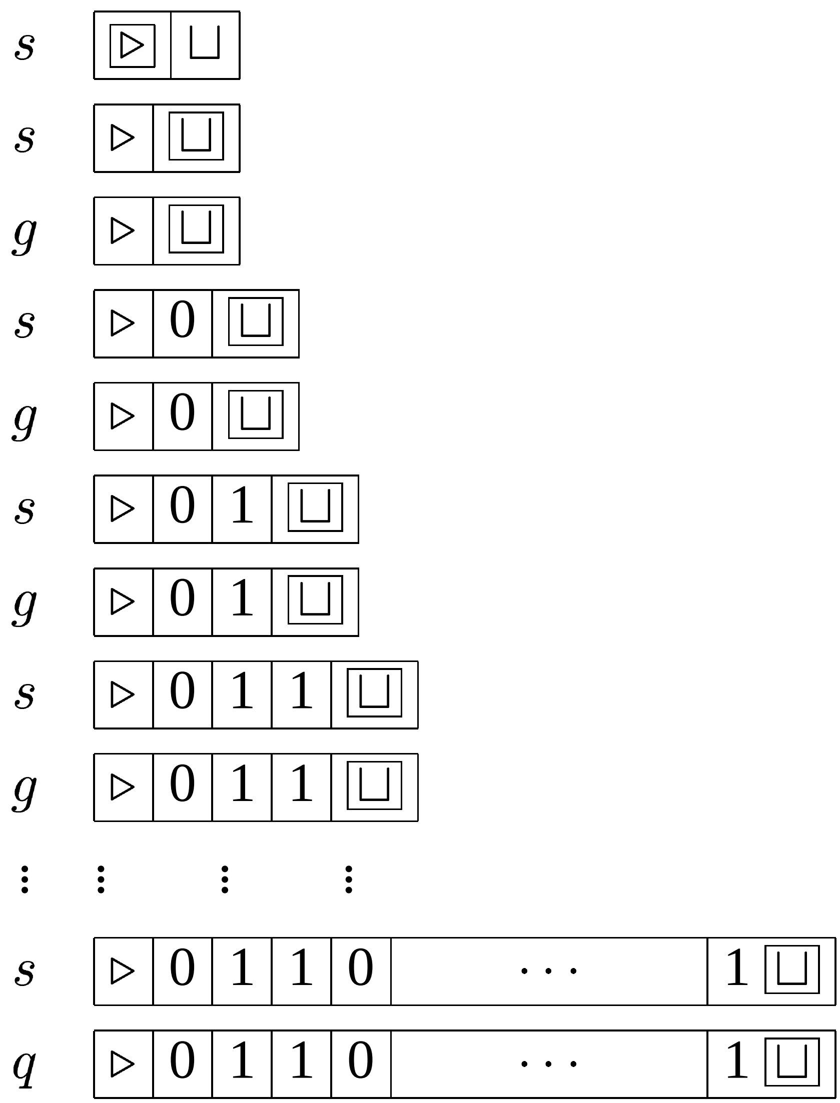
Each step takes at most $O((t(n))^2)$ steps to simulate. ♠

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

- 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.

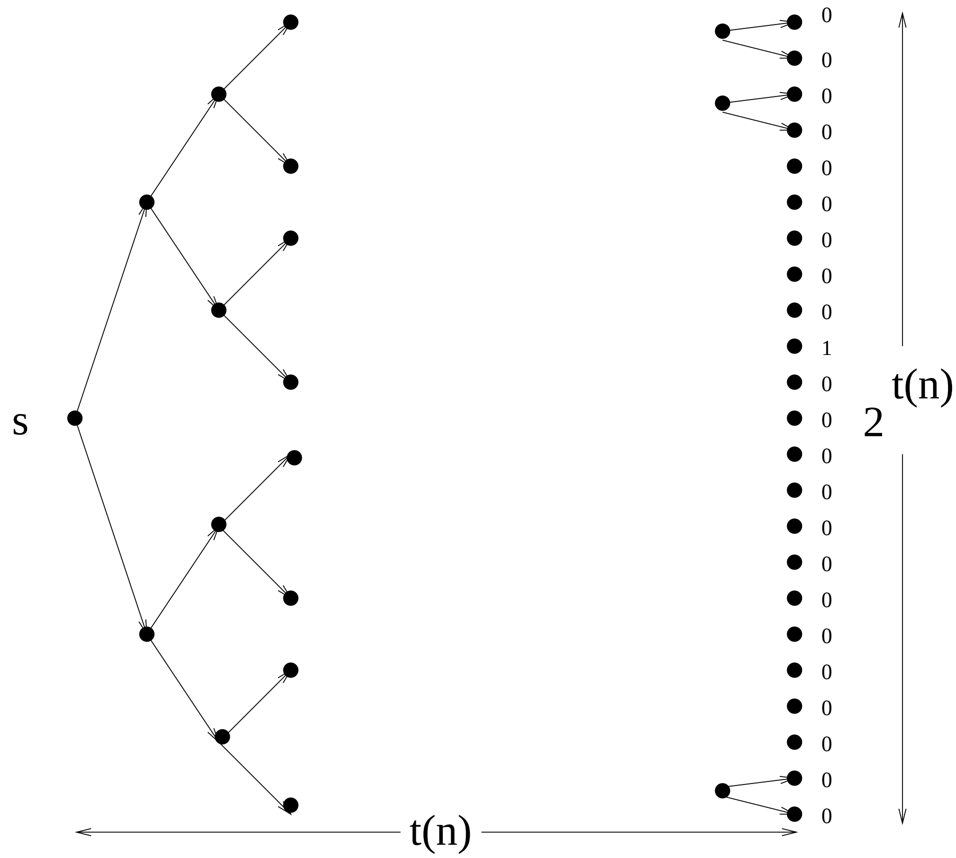
guess.tm	s	g	q
0			
1			
$\sqcup$	$g, \sqcup, -$ | $q, \sqcup, -$	$s, 0, \rightarrow$ | $s, 1, \rightarrow$	
$\triangleright$	$s, \triangleright, \rightarrow$		
comment	g or q	guess 0 or 1	the rest



Definition 5.14 The set accepted by a NTM, $N: \ \mathcal{L}(N) \equiv$

$$\{w \in \Sigma^\star \mid \text{some run of } N(w) \text{ halts with output "1"}\}$$

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. ♠



$$\textbf{NTIME}[t(n)] \;\equiv\; \text{probs. accepted by NTMs in time } O(t(n))$$

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

Theorem 5.15 *For any function* $t(n)$,

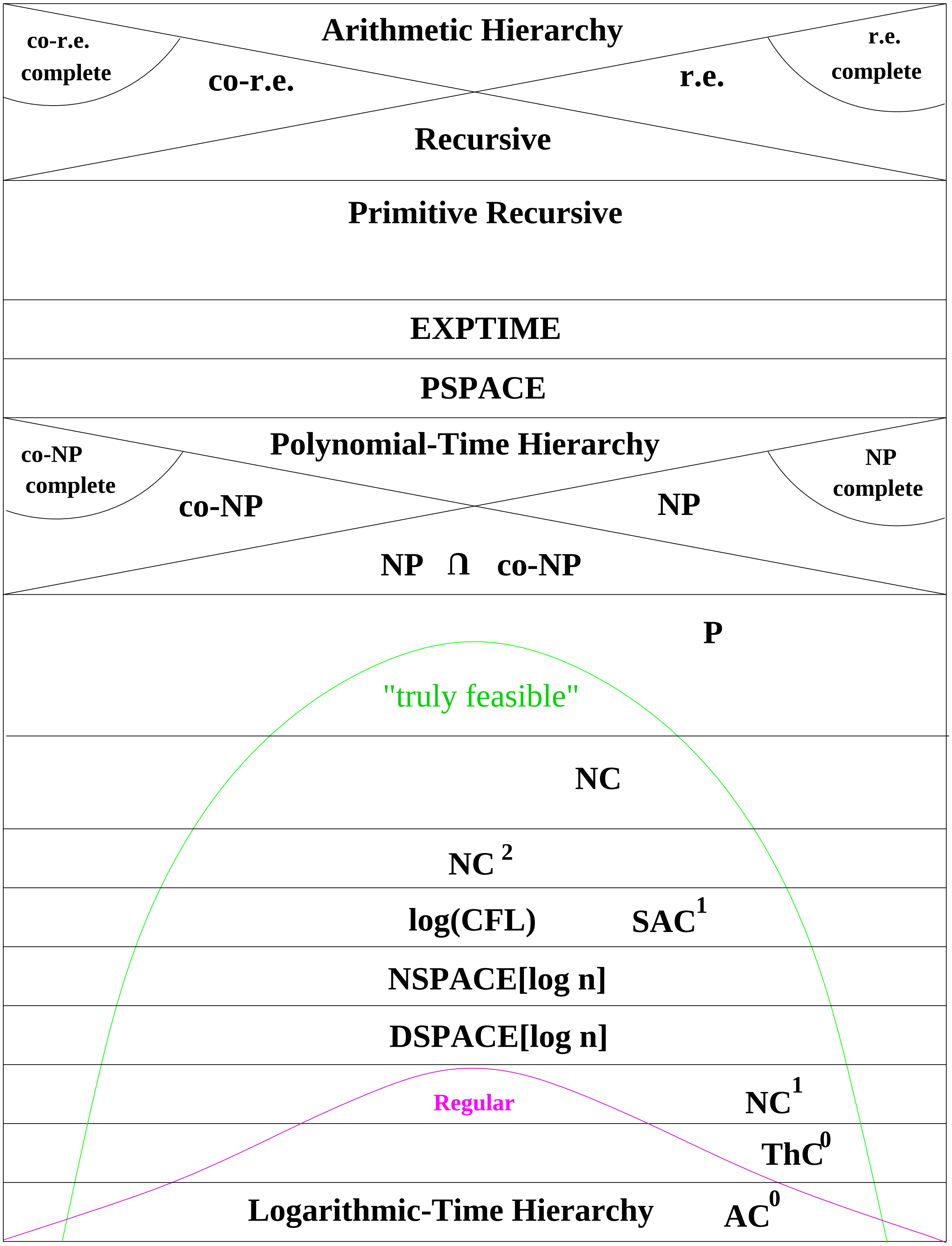
$$\textbf{DTIME}[t(n)] \;\subseteq\; \textbf{NTIME}[t(n)] \;\subseteq\; \textbf{DSPACE}[t(n)]$$

Recall: $\textbf{DSPACE}[t(n)] \;\subseteq\; \textbf{DTIME}[2^{O(t(n))}]$

Corollary 5.16

$$\textbf{L} \;\subseteq\; \textbf{P} \;\subseteq\; \textbf{NP} \;\subseteq\; \textbf{PSPACE}$$

Corollary 5.17 *The definition of* **Recursive** *and* **r.e.** *are unchanged if we use nondeterministic instead of deterministic Turing machines.*


Arithmetic Hierarchy
co-r.e. complete
co-r.e.
r.e.
r.e. complete
Recursive
Primitive Recursive
EXPTIME
PSPACE
Polynomial-Time Hierarchy
co-NP complete
co-NP
NP
NP complete
NP ∩ co-NP
P
"truly feasible"
NC
NC 2
log(CFL)
SAC1
NSPACE[log n]
DSPACE[log n]
Regular
NC1
ThC0
Logarithmic-Time Hierarchy
AC0