Definition 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 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.

(The input tape is considered “read-only” and not counted as space used.)

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

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

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

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

$M(w)$ has at most,

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

possible configurations.

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

Corollary $\mathbf{L} \subseteq \mathbf{P} \subseteq \mathbf{PSPACE}$

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

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

Theorem For any function $t(n)$,

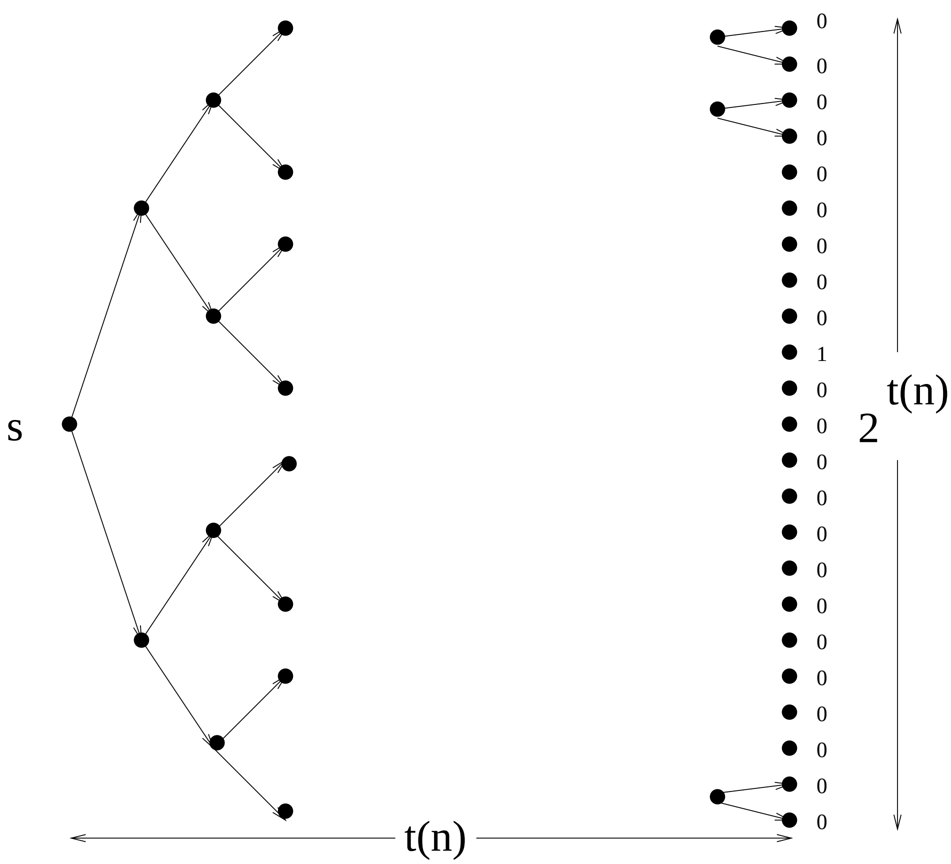
$$\textbf{DTIME}[t(n)] \subseteq \textbf{NTIME}[t(n)]$$

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

Corollary

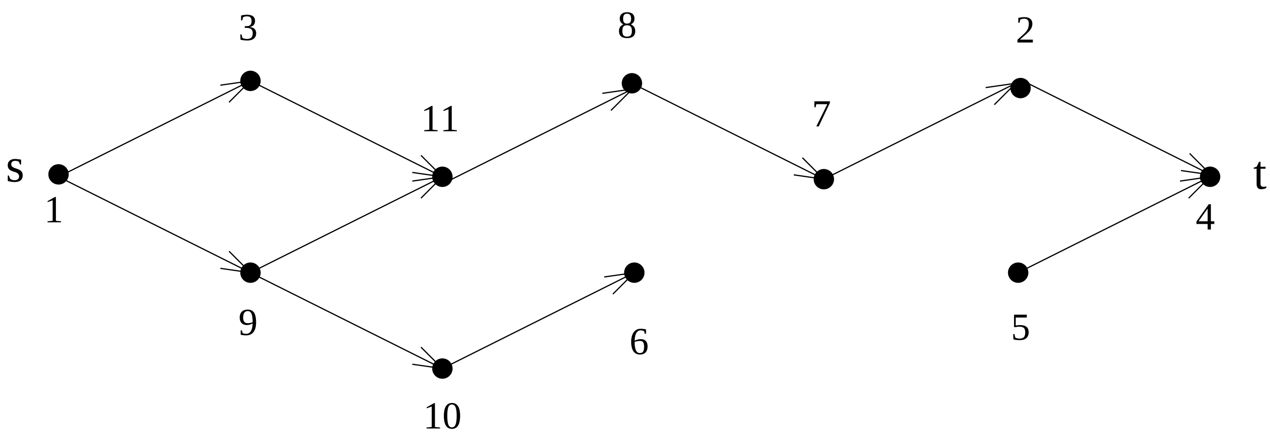
$$\textbf{L} \subseteq \textbf{P} \subseteq \textbf{NP} \subseteq \textbf{PSPACE}$$

NSPACE$[s(n)]$ is the set of problems accepted by NTMs using at most $O(s(n))$ space on each branch.



$s(n)$ space on each branch

Definition REACH is the set of directed graphs G such that there is a path in G from s to t.



Proposition REACH $\in$ **NL** $=$ **NSPACE**$[\log n]$

Proof: Let $n = |V|$, a ☐ b ☐

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

This algorithm accepts REACH in **NSPACE**$[\log n]$. ♠

Definition 16.1 Problem T is *complete* for complexity class $\mathcal{C}$ iff

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

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

Theorem 16.2 REACH *is complete for* **NL**.

Proof: Let $A \in$ **NL**, $A = \mathcal{L}(N)$, uses $c \log n$ bits of worktape.

Input w, $n = |w|$

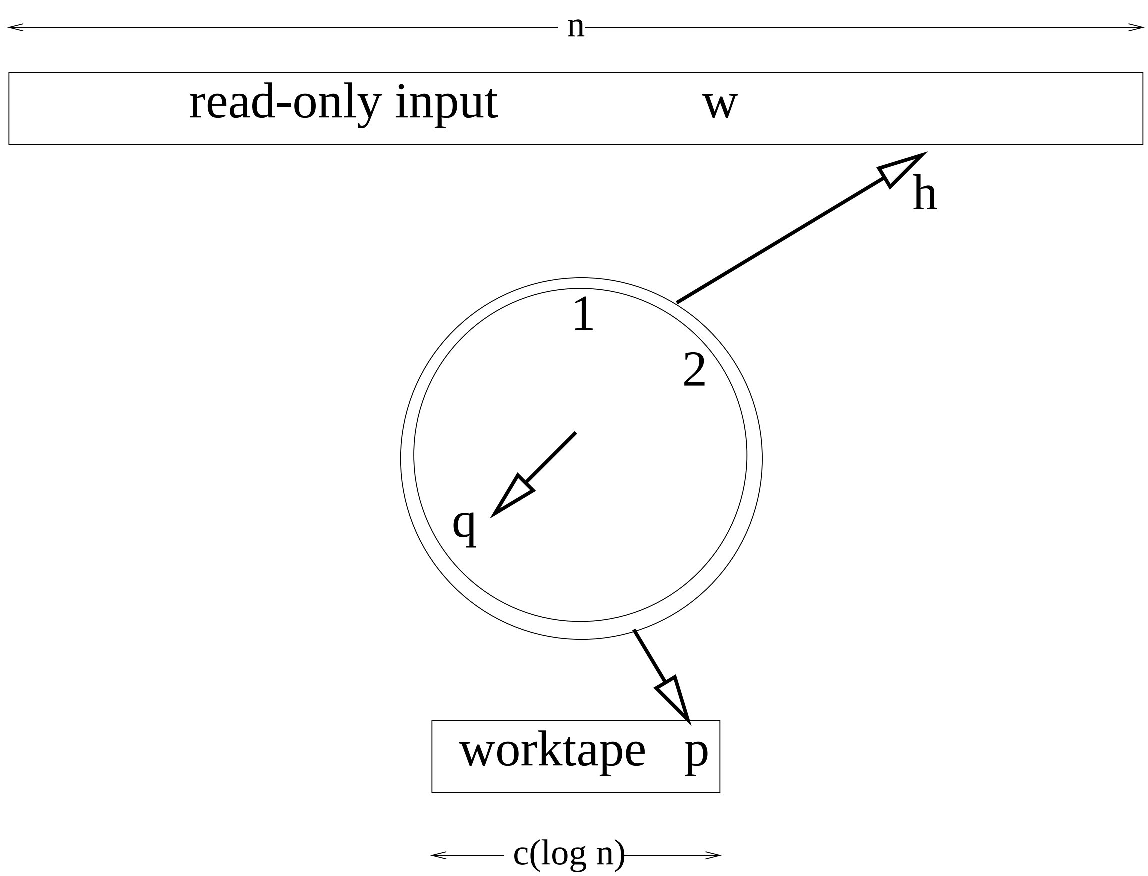
$$w \mapsto \text{CompGraph}(N, w) = (V, E, s, t)$$

$$V = \{\text{ID} = \langle q, h, p \rangle \mid q \in \text{States}(N), h \leq n, |p| \leq c\lceil \log n \rceil\}$$

$$E = \{(\text{ID}_1, \text{ID}_2) \mid \text{ID}_1(w) \underset{N}{\longrightarrow} \text{ID}_2(w)\}$$

$$s = \text{initial ID}$$

$$t = \text{accepting ID}$$



$$\text{CompGraph}(N, w) \quad = \quad (V, E, s, t)$$

$$V \;=\; \{\text{ID} = \langle q, h, p \rangle \;\mid\; q \in \text{States}(N), h \leq n, |p| \leq c\lceil \log n \rceil\}$$

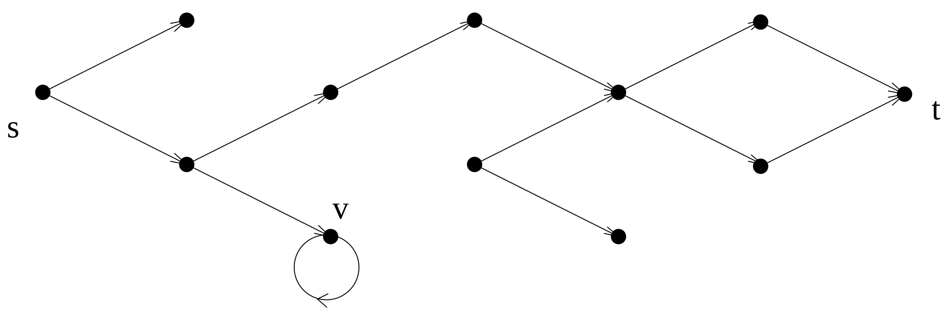
$$E \;=\; \{(\text{ID}_1, \text{ID}_2) \;\mid\; \text{ID}_1(w) \underset{N}{\longrightarrow} \text{ID}_2(w)\}$$

$$s \;=\; \text{initial ID}$$

$$t \;=\; \text{accepting ID}$$

Claim:

$$w \in A \;\Leftrightarrow\; w \in \mathcal{L}(N) \;\Leftrightarrow\; \text{CompGraph}(N, w) \in \text{REACH}$$



♠

Corollary 16.3

$$\textbf{NL} \quad \subseteq \quad \textbf{P}$$

Proof: REACH $\in$ **P**

P is closed under (logspace) reductions.

That is

$$(B \in \textbf{P} \quad \wedge \quad A \leq B) \quad \Rightarrow \quad A \in \textbf{P}$$ ♠

Theorem 16.4 *If $f(n)$ is a $\mathcal{C}$-constructible function; $\mathcal{C}$ is DSPACE, NSPACE, DTIME, or NTIME; and, if $g(n)$ is sufficiently smaller than $f(n)$ then $\mathcal{C}[g(n)]$ is strictly contained in $\mathcal{C}[f(n)]$.*

$g(n)$ sufficiently smaller $f(n)$ means

$\lim_{n\to\infty} \frac{g(n)}{f(n)} = 0$ $\qquad$ $\lim_{n\to\infty} \frac{g(n)\log(g(n))}{f(n)} = 0$

$\mathcal{C} = \textbf{DSPACE}, \textbf{NSPACE}, \textbf{NTIME}$ $\qquad$ $\mathcal{C} = \textbf{DTIME}$

Definition 16.5 Function $f : \mathbf{N} \to \mathbf{N}$ is $\mathcal{C}$**-constructible** if the map

$$1^n \mapsto f(n)$$

is computable in the complexity class $\mathcal{C}[f(n)]$. For example a function $f(n)$ is DSPACE-constructible if the function $f(n)$ can be deterministically computed from the input 1^n, using space at most $O[f(n)]$. ♠

Fact 16.6 *All reasonable functions greater than or equal to $\log n$ are DSPACE-constructible, and all reasonable functions greater than or equal to n are DTIME-constructible.*

Theorem 16.7 (Space Hierarchy Theorem)

Let $f \geq \log n$ be a space constructible function. If

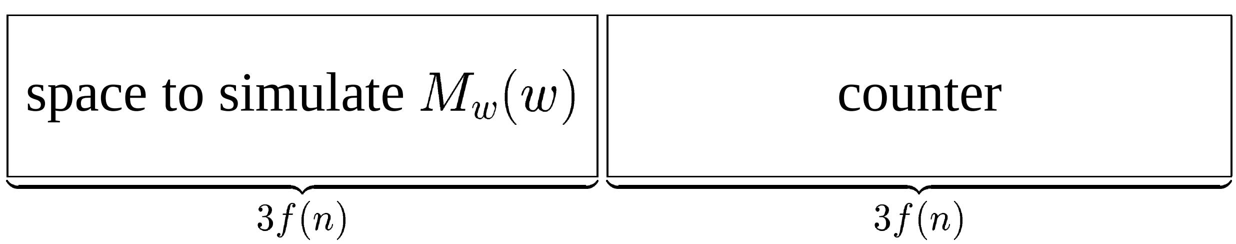
$$\lim_{n\to\infty} \frac{g(n)}{f(n)} = 0$$

Then, $\quad$ **DSPACE**$[g(n)] \quad \subsetneq \quad$ **DSPACE**$[f(n)]$.

Proof: Construct following **DSPACE**$[f(n)]$ machine, D:

Input: $w, \quad n = |w|$

1. Mark off $6f(n)$ tape cells, (f space constructible)
2. Simulate $M_w(w)$ using space $3f(n)$, time $\leq 2^{3f(n)}$
3. **if** ($M_w(w)$ needs more space or time) **then accept**
4. **else if** ($M_w(w)$ = **accept**) **then reject**
5. **else accept** // ($M_w(w)$ = **reject**)



Claim: $\mathcal{L}(D) \in \textbf{DSPACE}[f(n)] - \textbf{DSPACE}[g(n)]$

$\mathcal{L}(D) \in \textbf{DSPACE}[f(n)]$ by construction.

Suppose $\mathcal{L}(D) \in \textbf{DSPACE}[g(n)]$.

Let $\mathcal{L}(M_w) = \mathcal{L}(D)$, M_w uses $cg(n)$ space.

Choose N s.t. $(\forall n > N)(cg(n) < f(n))$.

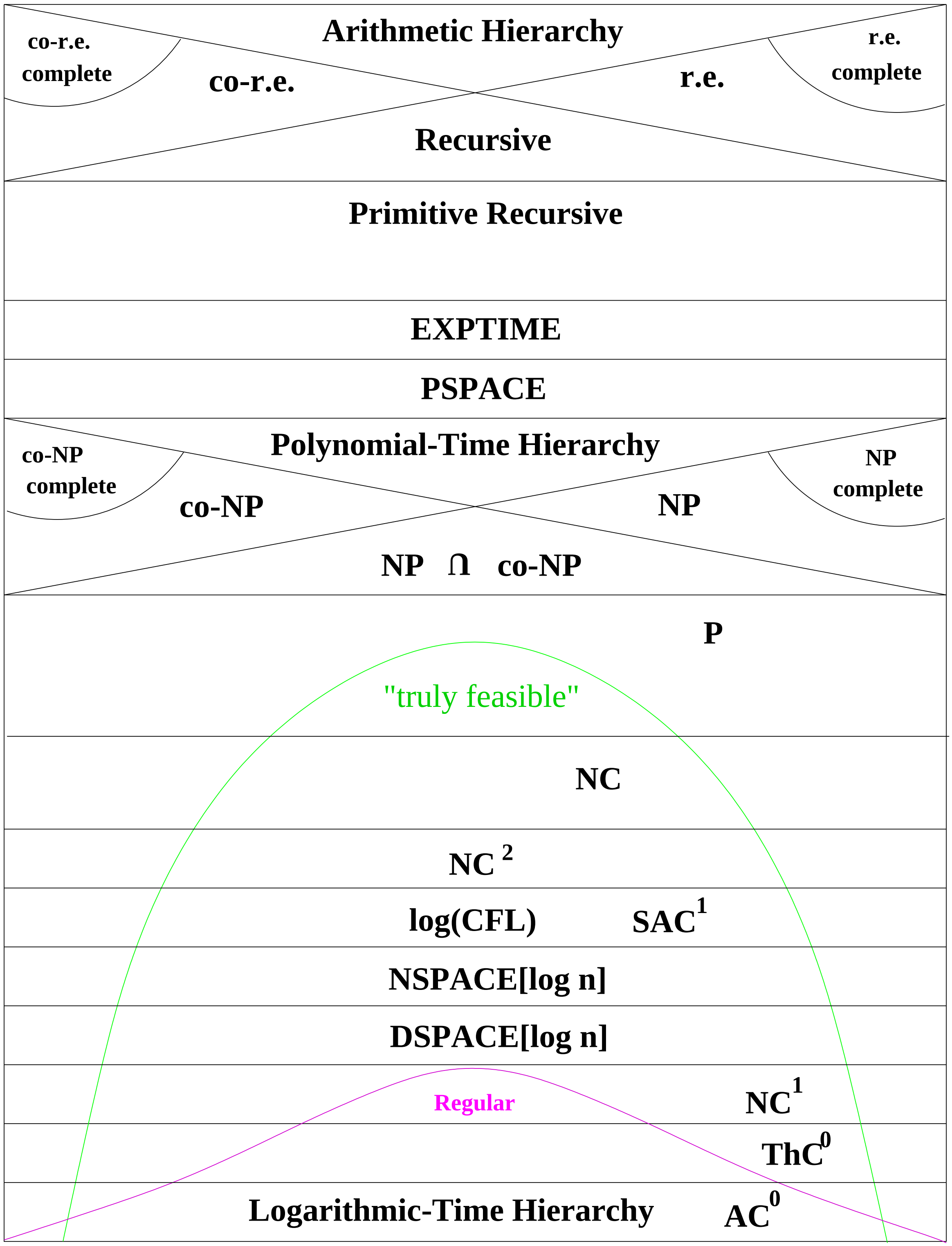
Choose w', $M_{w'}(\cdot) = M_w(\cdot)$, $|w'| > N$

On input w', D successfully simulates $M_{w'}(w')$ in $3f(n)$ space and $2^{3f(n)}$ time.

$$w' \in \mathcal{L}(D) \;\Leftrightarrow\; w' \notin \mathcal{L}(M_{w'}) \;\Leftrightarrow\; w' \notin \mathcal{L}(M_w) \;\Leftrightarrow\; w' \notin \mathcal{L}(D)$$

$$\Rightarrow\Leftarrow$$

♠


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

For any complexity class $\mathcal{C}$, define $F(\mathcal{C})$, the total, polynomially-bounded functions computable in $\mathcal{C}$ as follows:

$$F(\mathcal{C}) \;=\; \left\{ h : \Sigma^\star \to \Sigma^\star \;\middle|\; \begin{array}{c} (\exists k)(\forall x)(|h(x)| \le k|x|^k \\ \text{and} \quad \text{bit-graph}(h) \in \mathcal{C}) \end{array} \right\}$$

$$\text{bit-graph}(h) \;=\; \{\langle x, i, b\rangle \;|\; \text{bit } i \text{ of } h(x) \text{ is } b\}$$

Idea: $f \in F(\mathcal{C})$ iff

1. f is polynomially bounded, and,
2. bit i of $f(w)$ is uniformly computable in $\mathcal{C}$ and co-$\mathcal{C}$.